Development of a Classifier for Non-Stationary Disturbances in Power Systems

by

Mohammedi Raja

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY KANPUR

March, 2000

Development of a Classifier for Non-Stationary Disturbances in Power Systems

A thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Technology

in

Electrical Power Systems

by

Mohammedi Raja
1017047/9810438

under the guidance of

Prof Dr -Ing Jurgen Stenzel **Prof Prem Kumar Kalra**

Institut fur Elektrische Energieversorgung
Technische Universitat Darmstadt
Germany

Department of Electrical Engineering
Indian Institute of Technology Kanpur
India

February 21, 2000

CERTIFICATE

This is to certify that the thesis entitled "**DEVELOPMENT OF A CLASSIFIER FOR NON STATIONARY DISTURBANCES IN POWER SYSTEMS**" submitted by Mohammedi Samunbhai Raja to Indian Institute of Technology Kanpur for the award of Master of Technology in Electrical Engineering is a bonafide record of project work carried out under our supervision He is jointly guided by Prof Dr -Ing Jurgen Stenzel Technische Universitat Darmstadt, Germany and Prof Prem Kumar Kalra Indian Institute of Technology Kanpur India Contents of the thesis in full or in parts, have not been submitted to any other institute or university for the award of degree or diploma

Dr -Ing Jurgen Stenzel
Professor
Technische Universitat Darmstadt
Germany

Dr Prem Kumar Kalra
Professor
Department of Electrical Engg
Indian Institute of Technology
Kanpur India

INSTITUT FUR ELEKTRISCHE ENERGIEVERSORGUNG

TECHNISCHE UNIVERSITAT DARMSTADT

Prof Dr Ing J Stenzel

Certification

Raja, **Mohammedi Samunbhai** * 6 8 1976

Indian Institute of Technology Kanpur India (Prof Prem K Kalra)

Evaluation of the Master Thesis

Development of a Classifier of Non stationary Disturbances in Power Systems

Under my supervision and the guidance of Dipl Ing Heiko Englert Mr Raja prepared his thesis After being introduced into the subject he worked mainly independently and he was very diligent The effort in programming was very demanding The test results have been convincing The presentation and discussion of his thesis testified his good understanding of the problem Considering the various aspects of his thesis (research work results written thesis and presentation) my evaluation leads to the mark **very good**

(German marks 1 very good 2 good 3 satisfactory 4 sufficient 5 unsatisfactory)

During his stay in Germany Mr Raja had the opportunity to spend some time in industry

- **repas AEG Automation GmbH** Dreieich (2 weeks)
- **HEAG,** Darmstadt (local utility 2 weeks)

and to join an

- **Excursion to Austria** (Industry and Power Plants week)

organised by the Department of Electrical Engineering TU Darmstadt

Darmstadt 20 2 2000

J Stenzel

(Prof Dr Ing Jurgen Stenzel)

INSTITUT FUR ELEKTRISCHE ENERGIEVERSORGUNG

TECHNISCHE UNIVERSITAT DARMSTADT

Prof Dr Ing Gerd Balzer
Prof Dr -Ing Jurgen Stenzel

Darmstadt 08 02 2000

Diplomarbeit Nr 396

fur

Herrn Mohammedi Raja

Development of a classifier for non stationary disturbances in power systems

The power quality is increasingly affected by a large number of power electronic devices The analysis and assessment of short termed disturbances is usually done by a manual shifting and sorting The main goal is to develop a tool which classifies the disturbance waveforms automatically

The first step is to analyse the characteristics of various disturbance types This should lead to a feature selection representing the input for a classifier
Secondly the type of the classifier (Distance Functions Artificial Neuronal Networks Fuzzy Sets) should be determined This is done by the analysis of the patterns created by modelling and simulating various disturbances
Finally the choosen classifier should be implemented in MATLAB or C++ and the results should be discussed

Ausgabetermin	01 Juli 1999
Einlieferungstermin	01 Februar 2000
Betreuung	Dipl Ing Heiko Englert

J Stenzel

Prof Dr Ing J Stenzel

Acknowledgements

I am deeply indebted to Prof Prem kumar Kalra Indian Institute of Technology Kanpur for his permission to do M Tech thesis in Germany through DAAD student exchange program He also guided me by giving advices and providing some literature regarding my work I would like to express my deep gratitude to Prof Jürgen Stenzel Technische Universitat Darmstadt for his acceptance as my thesis supervisor in Germany He advised and encouraged well during my work I am thankful to Prof Gerd Balzer to advise and encourage me during my work I am really indebted to Mr Heiko Englert for his invaluable help and guidance regarding my thesis work as an adviser He helped me in different stages of my work from the beginning to the end I would like to thank Mr Francois Lorrain who helped to develop the *Graphical User Interface (GUI)* program in Matlab I am grateful to Mr Markus Brandl Mr Kristjan Halldorsson and other research fellows in my institute to help me regarding some literature and others

Abstract

The classification of disturbances of power systems is the important task in automated power quality assessment system This thesis work is mainly concentrated on the design of a classifier for disturbances in power systems It uses characteristic features of disturbances to design and evaluation of the classification system The simulation of the classifier is done with artificially generated data of disturbances using known ranges of various disturbances features Various classification techniques like probabilistic fuzzy neural network and geometric are tested to design the suitable classifier for power quality disturbances classification The suggested classifier uses parallel classification structure of three selected classifiers The requirements of the classifier include assessment of the type of disturbance quality of classification and adaptability to new unknown disturbance The sequential classification approach is also implemented for superimposed disturbances classification

Contents

1 Introduction

PQ has ever been a topic of considerations in power systems but increasing attention is been given to PQ since recent years. The interest in quality of power involves all three parties concerned with power business: utility companies, equipment manufacturers and electric power consumers. Many reasons are responsible for the growing concern with PQ:

- the end user equipment has become very sensitive to PQ as a result of the wide range of microprocessor based applications,
- complexity and interconnection of industrial processes (a restart up of processes is costly),
- development and application of sophisticated power electronics (these devices are both source and victim of PQ disturbances),
- deregulation of the power market (PQ as a product feature),
- business competition causes rationalisation (reliability of electric power and as well PQ decreases).

These are convincing reasons to monitor and assess the PQ in power systems. So far the PQ diagnosis is very time consuming, because of the large amount of recorded data and its manual analysis. So an automatic technique of PQ analysis is desirable.
The project "Development of a System of Automated Classification and Assessment of PQ Disturbances" pursues the approach to identify the disturbances by their characteristic features. With information about type, location and statistical parameters (frequency of occurrence, etc.) the analysis of the PQ disturbances is to be realised. For this purpose methods of pattern classification, an application of artificial intelligence, are used. The goal of the present thesis is the implementation of pattern classification techniques to the PQ problem. The choice of a suitable classifier represents the main focus. In the following chapter the principles of pattern classification and of the development of a pattern classification system are introduced. In Chapter 3 the application of pattern classification to PQ is described. Afterwards the studied classifiers are presented. The methods

for classifier test and the test results are given in chapter 5 The selected classifier system and its realisation in Matlab is shown in chapter 6 Finally the results of this thesis and new proposals are summarised

2 Classification Basics

The goal of pattern recognition systems is the classification of objects (patterns) into a number of categories or classes. Pattern recognition has a lot of practical applications, e.g. probably the most popular one is character recognition. Here, the patterns are described by pixels of digitized character images which are mapped to classes – the letters of the alphabet.

2.1 Principles of Pattern Classification

In the preceding example pixels of images are used to represent patterns. Such measurable qualities of patterns are called features. In general case n features x_i, $i = 1, 2, \ldots, n$ are used and form the feature vector

$$\mathrm{x} = [x_1, x_2, \ldots, x_n]^T \tag{2.1}$$

Graphically it can be shown like in Figure 2.1. Each of the feature vector

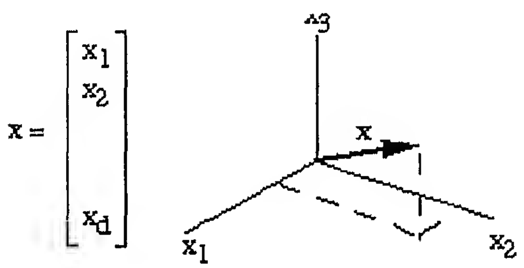


Figure 2.1: Feature vector representation[2]

describes uniquely a single pattern (object). The patterns are assigned to a finite set of classes

$$\Omega = \{\omega_1, \omega_2, \ldots, \omega_k\} \tag{2.2}$$

Where k is the number of classes
In mathematical sense the mapping of classification from feature space to decision space is stated as follows

$$S \quad \mathrm{x} \to \Omega \tag{2 3}$$

To illustrate the classification task the feature space spanned by two dimensional feature vectors is shown in Fig 2 2
The circles and crosses represent feature vectors of sample patterns of two differ

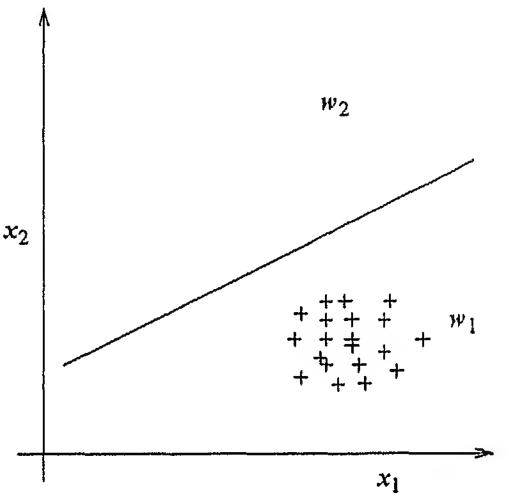
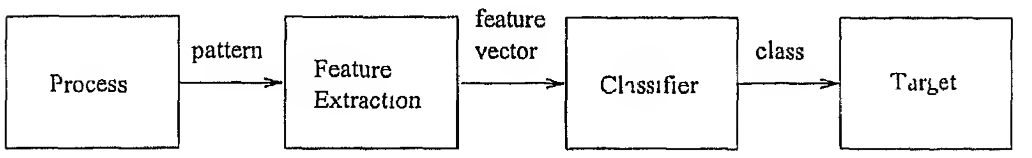


Figure 2 2 Two dimensional feature space

ent categories The straight line is known as the decision line which constitutes the classifier It's role is to divide the feature space into regions which belongs to either class ω_1 or class ω_2 If the feature vector of an unknown pattern falls into the region of ω_1, it is classified as class ω_1 But this doesn't implicate that the decision is correct If not, a misclassification has occurred
How the whole pattern classification system is embedded in a real world scheme is depicted in Fig 2 3
Generally a pattern classification system consists of the following three steps

i) **Feature Extraction** Some features that can express the patterns well are extracted from the patterns in the real world The features are usually expressed as numerical values The patterns will be classified in this feature space

ii) **Construction of a Classifier and Testing** A classifier is constructed on the basis of training samples in the feature space In this step, the number of



Pattern Classification System

Figure 2 3 Generalised structure of pattern classification system

the training samples are usually limited In testing stage of the classifier the performance of the classifier can be judged on the basis of total number of misclassified samples out of total known test samples of a class Usually, test samples are other than training samples

iii) Classification of Unknown Samples Unknown samples are classified using the classifier When the classifier does not have a sufficient classification power, many unknown samples are misclassified!

2 2 Design of Pattern Classification Systems

The design of a pattern classification system for a given classifier task can be divided into the following stages

1 Determination of patterns and classes

2 Feature generation

3 Feature selection

4 Classifier design

5 System evaluation

Figure 2 4 shows the various stages As it can seen from the feedback rows, these stages are not independent The stages are interrelated and to improve the overall performance of the system, one may go back to redesign earlier stages It is obvious that this classifier system design is a difficult optimisation process To facilitate an optimal design there exist some fundamental rules for each stage, which are pointed out as follows

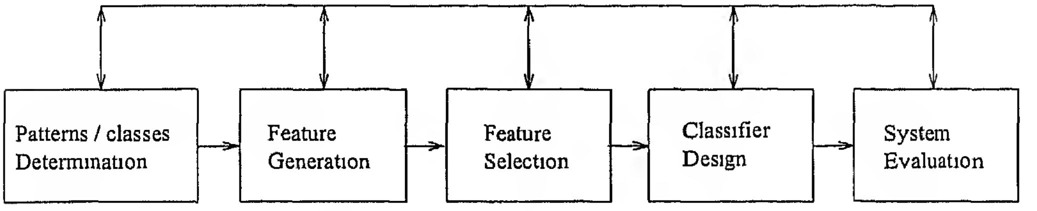


Figure 2 4 The basic stages involved in the design of a classification system

STAGE I Determination of patterns and classes

At the first stage the designer has to analyse the process, on which a classifier system is to be applied He has to ask the questions what are the possible events of the process and how many events are there? It is important to notice, that at this early stage the structure of the classifier is set For a simple structure of an individual classifier the classes should represent unique events of the process This means, that the classes are exclusive In contrary when classes coincide or when classes and sub classes come into question as result, a hierarchical classifier structure is recommended It is often helpful to define a class, that represents the normal condition of the process This relieves the performance tuning of the classifier

STAGE II Feature Generation

Here, the designer asks for the typical patterns of the process, and which could be mapped to the classes It is essential to use significant patterns of the process From this patterns, those features are to be extracted, which are measurable and can be represented numerically (as number or boolean) These features combined to a feature vector form a sample vector, which could be mapped to a certain class Every feature has a range of measurement called feature range which has subranges for different disturbances Every feature range differs from another feature range by it's units So, it is required to normalise all feature ranges on the same scale, e g 0 100 % or 0 1 pu, before classification For an excellent performance of a classifier system a high number of sample vectors are requested But there are cases, where the process only provides a small rate of samples or the classifier is to be designed on the fly In that case modeling the process and generation of artificial feature samples is suggested This simulated data offers some advantages

- cheaper than real data,
- easy and fast to generate,

- more flexible, in respect to feature ranges and distributions

STAGE III Feature Selection

Often the process provides a lot of features, that describe a certain class Out of them those features are to select, which offer the most significant characteristics and allow optimal class separability Considering the feature space the requirement for optimal class separability is pictured by Fig 2 5
Fig(c) shows the best (b) the worst and (a) a moderate separability Another

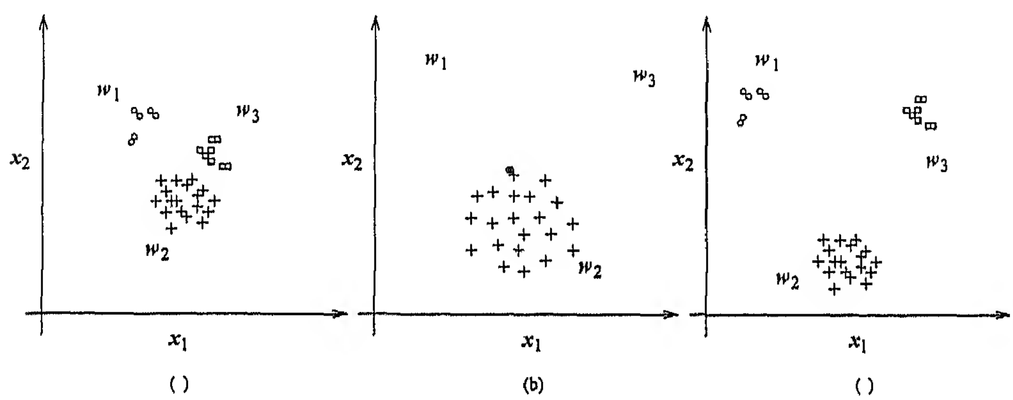


Figure 2 5 Classes with (a) small within class variance and small between class distances, (b) large within class variance and small between class distances and (c) small within class variance and large between-class distances

consequence is, that a high number of features means the dimension of the resulting feature vector is high For fast classification a low complexity of the feature space is advantageous This implicates, that the features is to be analysed for redundant information Generally linear dependencies of features should be avoided The example of three features illustrates this height, length, surface Here either surface or the two other features are redundant This example offers another problem which one is more redundant than the others? Here the principal component analysis (PCA) is a useful tool to identify linear dependencies of different features (more about that in chapter 3)

STAGE IV Classifier Design

The classifier itself is the core of the pattern classification system and embodies the artificial intelligence of the whole system The classifier realises the link between the feature vectors and the classes It decides to which event an occurring

pattern is assigned to The decision bases on the comparison of unknown with known data As a consequence the classifier is to make known with possible data earlier In terms of pattern recognition the classifier has to be trained with *a priori* information Here it has to be distinguished between supervised and unsupervised learning In the first case of supervised learning for each sample vector the right appropriate class is known In case of unsupervised learning the information of class membership is missing Here the goal is to unravel the underlying similarities of the sample vectors and cluster (group) similar vectors together In this work we only deal with supervised pattern recognition so the interested reader is referred to additional literature[2] Basically the classifiers can be divided into three types according to their mathematical background

- linear classifiers (e g Euclidean distance classifier)
- nonlinear classifiers (e g ANN Fuzzy and polynomial classifier)
- probabilistic classifiers (Bayes classifier)

It is hardly possible to decide which type of classifier is suited best for a given application (it is the main focus of this work to find the best classifier for the PQ problem) The decision depends on a lot of factors, which are regarded in chapter 4

STAGE V System Evaluation

When a certain classifier is chosen it has to be trained The number of the training sets (set = sample vector + class information) has a large influence on the performance of the classifier Here the question is how many training sets are necessary The answer is simple, but nevertheless unprecise the more, the better To test the classifier it has to be confronted with unknown data (Generalisation) Then the outputs of the classifier are compared to the real class membership of the test sets In case of poor results when a large percentage of the outputs and *a priori* information mismatches improvements have to be made Figure 2 4 states, that the optimisation process is possible at any stage So any change in a stage should be followed by a system evaluation until the results are satisfactory

3 Adaptation to PQ problems

We have to know the various characteristics of different events in power system and their interrelation before determination of classes For the events occurring simultaneously (mixed or dependent events), it is required to design the classifier for more than one stage classification Determination of relation between power system events and various features is one of the important tasks for their selection Information about range of feature values is required before generation of feature vectors of different classes

3 1 Structure of PQ-Classification System

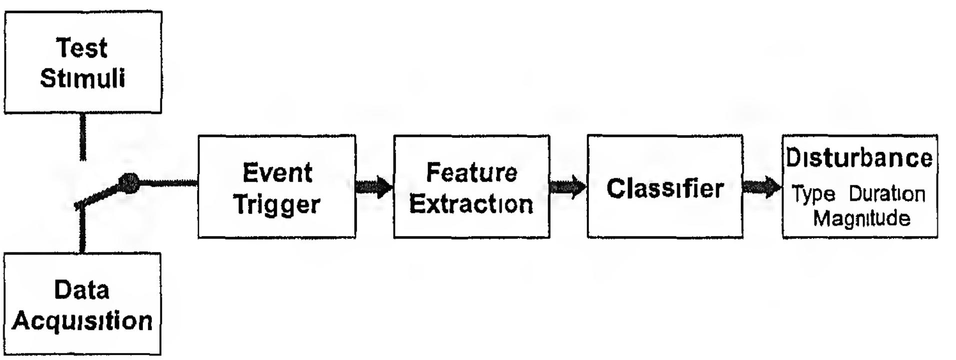


Figure 3 1 Generalised structure of PQ Classification system

PQ assessment is a sequence of five steps as shown in Figure 3 1 Data acquisition is an interface between power line measurement system and PQ analysis platform Event trigger recognises when there is a new event in the line and provides the event waveform pattern to next stage In feature extraction stage sufficient features from the disturbance pattern are extracted Classifier detects the disturbance type from feature vector PQ of the event can be judged with added information of duration and magnitude of the event and rate of occurrence along with disturbance type The main concentration in this thesis work is the

classification of PQ disturbances In the following sections we discuss some steps about the system design for the PQ classification problem

3 2 Determination of Patterns and Classes

Analysis of PQ problem provides events called disturbances Every disturbance s information is available in terms of a specific waveform, e g voltage waveform over a time range, called pattern This pattern could be transformed into in formation of different signals, e g peak value RMS (Root Mean Square) value duration rise time THD (Total Harmonic Distortion) value Dominant Inter harmonics (DIH) etc create a vector of these features called a feature vector Different disturbances differ by their special characteristics of various features Disturbances which differ by various ranges of the same discriminating features can be considered for classifier design e g impulsive transient, oscillatory tran sient, voltage sag, voltage swell, over voltage, under voltage need information about V_m, V_{RMS}, t , T_e and DIH As mentioned in STAGE I of Chapter 2 2 every disturbance occurring exclusively defines one class Feature vectors (patterns) of the same kind of disturbance is assigned to the same class Feature vectors having mixed characteristics of two independent disturbances can be clas sified by the same classifier structure without considering their special classes by providing some more features to the classifier In that case, one class is necessary which defines the normal event meaning there is no disturbance for reference More about this will be discussed in Chapter 4 Selection of classes also limited due to information available after feature extraction stage of classification system e g voltage unbalance needs information about the voltages of all three phases at the same time If the patterns don't provide those information then it is not possible to classify that disturbances

The major types of *non stationary* power system disturbances which persist for a certain duration only, related to PQ diagnostics are

- Impulsive transients
- Repetitive impulsive transients
- Oscillatory transients
- Voltage swells (surges)
- Voltage sags (dips)
- Overvoltages
- Undervoltages
- Short interruptions

- Sustained interruptions
- Voltage fluctuation (flicker)
- Voltage imbalance
- Power frequency variations
- DC offset (distortion)
- Harmonics (distortion)
- Notching (distortion)
- Interharmonics (distortion)
- Noise (distortion)

3 3 Features of PQ-Events

The PQ events are characterised by the analysis of the disturbance signals In the following, possible features are listed

Peak voltage V_m It is a peak(maximum) voltage measured within observation window It would be a peak value of impulse if there is otherwise it is equal to peak value of sinusoidal voltage waveform under normal operation

Phase angle placement of the event on the sine wave It is a angle between positive zero crossing of fundamental component and start of disturbance It is particularly important for transients and notching

Voltage magnitude V_{RMS} It is a RMS value of the voltage measured

rise time t_r As per standard definition of Rise Time of a waveform or impulse, it is a time from 10% to 90% of the front side of waveform or impulse One can replace this feature by Rate of Rise

Decay time t_d As per standard definition it is a time from 100% to 50% of the tail side of the waveform Rate of Decay is also one alternative for Decay Time

Duration of event T_e It is time period during which the disturbance is found in the waveform It varies from milliseconds to steady state, depending on the type of disturbance

Frequency spectrum It is an analysis of the waveform for the large range of frequencies

Dominant interharmonic The frequency of that interharmonic with the highest RMS voltage besides the fundamental voltage

Notch depth

Frequency of occurrence It can be measured on the basis of number of events the same disturbance occurred in one

Notch area

Percentage odd/even harmonics

Total harmonic distortion (THD) The definition of THD is given as follows

$$V_{THD} = \frac{\sqrt{\sum_{j=2}^{\infty} V_j^2}}{V_1}$$

where V_{THD} RMS voltage due to of all harmonic components V_j is a RMS value of h^{th} harmonic voltage and V_1 is RMS voltage of fundamental

3 4 Feature Selection for PQ-problem

For feature selection for PQ problem typical ranges of power system disturbances for some features are calculated on the basis of Table 3 1[3] Most known power system disturbances are categorised on the basis of their characteristics The basic characteristics are spectral content, duration of the phenomena and voltage magnitude (RMS) On the basis of these characteristics we could have some common features in which they differ by range in which they occur

3 4 1 Determination of distinct features

On the basis of typical characteristics known of disturbances, various features can be considered for classification of disturbances Selection of features depends on the disturbances to classify It is required to know the most discriminative feature for a particular disturbance Using reasoning and the knowledge of feature ranges from Table 3 1, the dependency table is developed as Table 3 2, in which the relation bond of the disturbance with different features is categorised in three relations It shows that the features must be selected with high relation for a particular feature Features having medium should be considered and and features with low dependency need not to be considered because they don t

Categories	Spectral	T_e	V_{RMS}
Transients			
Impulsive			
Nanosecond	5 ns rise	<50 ns	
Microsecond	1 μs rise	50 ns 1 ms	
Millisecond	0 1 ms rise	>1ms	
Oscillatory			
Low Freq	< 5 kHz	0 3 50 ms	0 4 per unit
Medium Freq	5 500 kHz	20 μs	0 8 per unit
High Freq	0 5 5 MHz	5 μs	0 4 per unit
Short Duration Var			
Instantaneous			
Sag		0 5 30 cycles	0 1 0 9 per unit
Swell		0 5 30 cycles	1 1 1 8 per unit
Momentary			
Interruption		0 5 cycles 3 sec	< 0 1 per unit
Sag		30 cycles 3 sec	0 1 0 9 per unit
Swell		30 cycles 3 sec	1 1 1 4 per unit
Temporary			
Interruption		3 sec 1 min	< 0 1 per unit
Sag		3 sec 1 min	0 1 0 9 per unit
Swell		3 sec 1 min	1 1 1 2 per unit
Long Duration Var			
Interruption		> 1 minute	0 0 per unit
Undervoltage		> 1 minute	0 8 0 9 per unit
Overvoltage		> 1 minute	1 1 1 2 per unit
Waveform Distortion			
Harmonics	0 100th H	steady state	0 20%
Inter harmonics	0 6 kHz	steady state	0 2%
Noise	broad band	steady state	0 1%
Voltage Fluctuations	< 25 Hz	intermittent	0 1 7%
Power Freq Var		< 10 s	

Table 3 1 Typical Characteristics of Power Disturbances[3]

provide any useful information about that particular disturbance In Table 3 3, some disturbances are shown with their typical feature ranges Here, features should be selected such that the disturbances can be distinguished by the value of features only Feature having same values for all disturbances should not be considered for classification For any disturbance, at least one feature must be

	Peak voltage	Voltage (RMS)	Duration of event	Rise time	Decay time	Frequency spec (THD)	Dominant I H	Notch area	Notch depth	System frequency
Impulsive transient	H	L	M	H	H	H	L			
Oscillatory transient	H	M	M	H	H	H	H			
Voltage sag	L	H	H	M	M					
Voltage swell	L	H	H	M	M					
Interruption		H	H	M	M					
Undervoltage	L	H	H	M	M					
Overvoltage	L	H	H	M	M					
Flicker	M	M	H			H				
Inter harmonics			H			H	H			
Harmonics			H			H	H			
Frequency variation			H							H
Notching	M	L	H			H		H	H	
DC-offset	H	H	H							
Noise	H		H			H				

Note **H** High relation or dependency, **M** Medium relation **L** Low relation

Table 3 2 Feature dependency of disturbances

there by which the disturbance differs from all other disturbances If new distur bances to classify is added then the new feature may be included It depends on the new disturbance characteristics More number of features required for more number of disturbances to classify

	V_m (pu)	V_{RMS} (pu)	T ms	t ms	THD %	DIH $\in \{0\ 1\}$	$\frac{\Delta f}{\Delta f_m}$ %
Impulsive Transient	2 5	1 15	05 5	01 2	0 1	0	0 10
Oscillatory Transient	2 5	1 4	05 10	01 5	0 25	1	0 10
V Sag	0 09	0 09	5 100	01 20	0 1	0	0 10
V Swell	11 14	11 14	5 100	01 10	0 1	0	0 10
Harmonics	09 11	09 11	5 100	01 10	1 5	0	0 10
Inter harmonics	09 11	09 11	5 100	01 10	0 1	1	0 10
Frequency Variation	09 11	09 11	5 100	01 10	0 1	0	10 100
No disturbance	09 11	09 11	5 100	01 10	0 1	0	0 10

Note

V_m	Peak voltage	V_{RMS}	Voltage magnitude
T	Duration of Event	t	Rise time
THD	Total Harmonic Distortion	DIH	Dominant Interharmonics
$\frac{\Delta f}{\Delta f_m}$	Change in frequency	Δf_m	$\lvert f_n - f_n \rvert = 1$ Hz $f_n = 50$ Hz

Table 3 3 Feature ranges of disturbances

3 4 2 Normalisation of feature intervals

The feature intervals normally differ with disturbance It is assessed from real world's disturbances observations and definitions Every disturbance feature has distinct interval The representation of the features could be done in percent (0 100%) or in per unit (0 1 pu) Some features are with their original range and their representation after normalisation to 100 % are shown in Table 3 4 Mathematically, normalisation of the feature range of peak voltage feature of Impulsive transient is stated as follows

$$x_{normalised} = \frac{x_{original}}{5pu} 100\% \qquad (3\ 1)$$

For example, there are some practical limitations for rise time such that it can be measured upto minimum value of 0 1 ms only since sampling rate is 20 kHz So

we have limitations to some disturbances to classify Here we concentrate only on transients and short duration specially instantaneous variations Normally it is represented by linear scaling in a standard unit The units of different features are not same Some are per unit values, some are percentage and some are other In feature space it is important to have symmetrical size of features in all dimensions because we provide the same weights to all features Really the importance of the feature range representation varies with type of classifier The scaling could be done on linear basis or logarithmic basis when features are represented in feature space The logarithmic scaling might be important for some features like rise time and duration in which the values varies from fraction of milliseconds to steady state Some uniformity of the size of feature vector group of every disturbance in feature space is required for better classifier performance

Feature	Total Feature Range	Representation
Peak Voltage	0 5 per unit	0 100%
Voltage (RMS)	0 5 per unit	0 100%
Duration	0 100 ms	0 100%
Rise time	0 10 ms	0 100%
Freq spectrum (THD)	0 5%	0 100%
Dominant Interharmonics	0 10 kHz	0 *or* 100%
System Frequency Variation	49 51 Hz	0 100%

Table 3 4 Features ranges and representation

3 4 3 Analysis of feature space

Power System disturbances can be represented by a vector in a n dimensional space, where n is number of features Every disturbance vector differs from an other disturbance vector by its location in the feature space Group of disturbance vectors of same type creates a region (cluster) in the feature space where those data points lie Every region of same type of disturbance vectors differs from another by its size and location The shape of the region depends on the data distribution Normally, the regions of different kind of disturbances don't overlap In figures 3 2 and 3 3, the representation of data points of different disturbances is shown in two dimensional and three dimensional feature space respectively The six disturbances considered are indicated in above figures The 6 features are selected with high relation using information from Table 3 2 are peak voltage RMS voltage, rise time duration, THD and Dominant Interharmonics

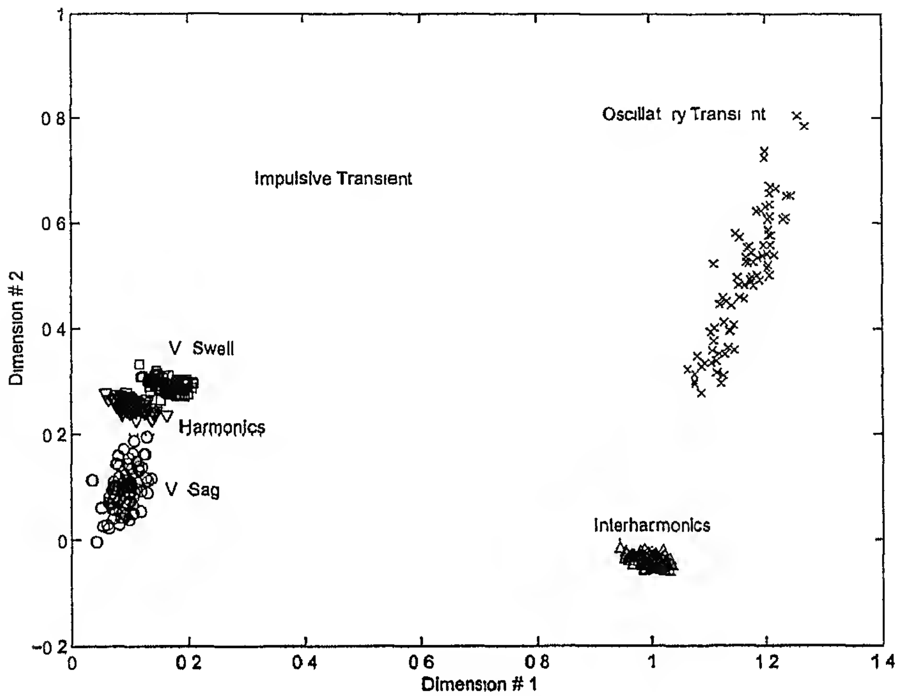


Figure 3 2 2 Dimensional representation of data points

3 4 4 Reduction of dimensionality

Principal Component Analysis (PCA) is used here to remove the redundant features from the dataset before training the classifier It is good to use data with less features because it becomes easy for the classifier to train, simplifies the problem and requires less memory PCA reduces the dimensionality of the data PCA also provides the eigenvalues in all principal components and on the basis of that one can decide how many minimum features are required to describe all most all information about the original dataset If some eigenvalues are very small compared to other eigenvalues then one can consider those number of features as redundant which is equal to number of eigenvalues are very small compared One has to find the features which are redundant by training and testing the classifier with reduction of features adapting trial and error method because which features are redundant can't be identified using PCA

The example shown in figure 3 4 explains reduction of redundant dimension technique One can judge how many minimum dimensions are necessary to separate the classes data points from one another

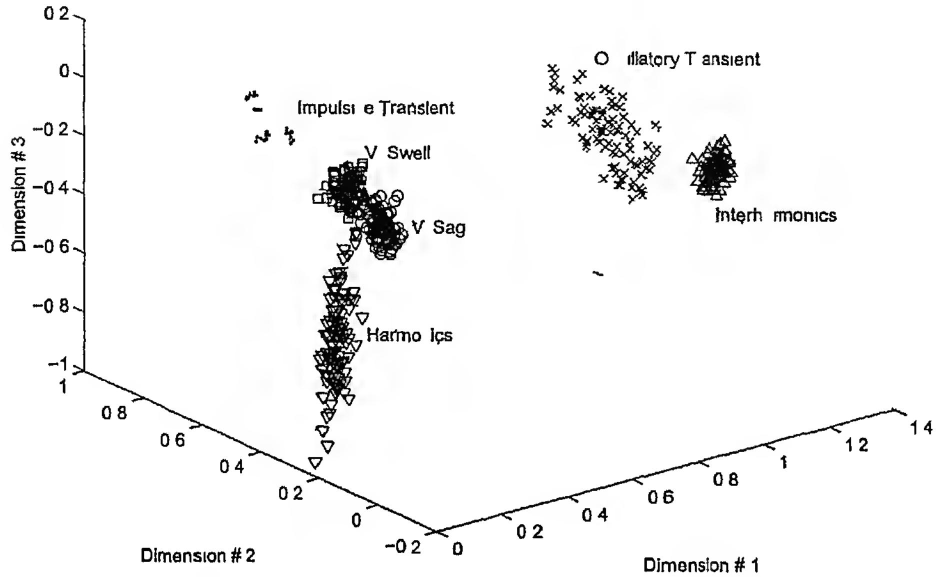


Figure 3 3 3 Dimensional representation of data points

Explanation of PCA

This section describes unsupervised Hebbian learning in a simple network setting to extract the m principal directions of a given set of data i e the leading eigenvector directions of the input vectors autocorrelation matrix
Principal component analysis (PCA) is equivalent to maximising the information content in the output of a network of linear units The aim of PCA is to extract m normalized orthogonal vectors $\mathbf{u}$, $i = 1, 2,$, m, in the input space that account for as much of the data s variance as possible Subsequently, the n dimensional input data (vectors $\mathbf{x}$) may be transformed to a lower m dimensional space without losing essential intrinsic information This can be done by projecting the input vectors onto the m dimensional subspace spanned by the extracted orthogonal vectors $\mathbf{u}$ according to the inner products $\mathbf{x}^T\mathbf{u}$ Since m is smaller than n, a dimensionality reduction of the data is achieved This, in turn, makes subsequent processing of the data (e g clustering or classification) much easier to handle
The following is an outline for a *direct* optimisation based method for determining the $\mathbf{u}_i$ vectors Let $\mathbf{x} \in \mathbf{R}^n$ be an input vector generated according to a

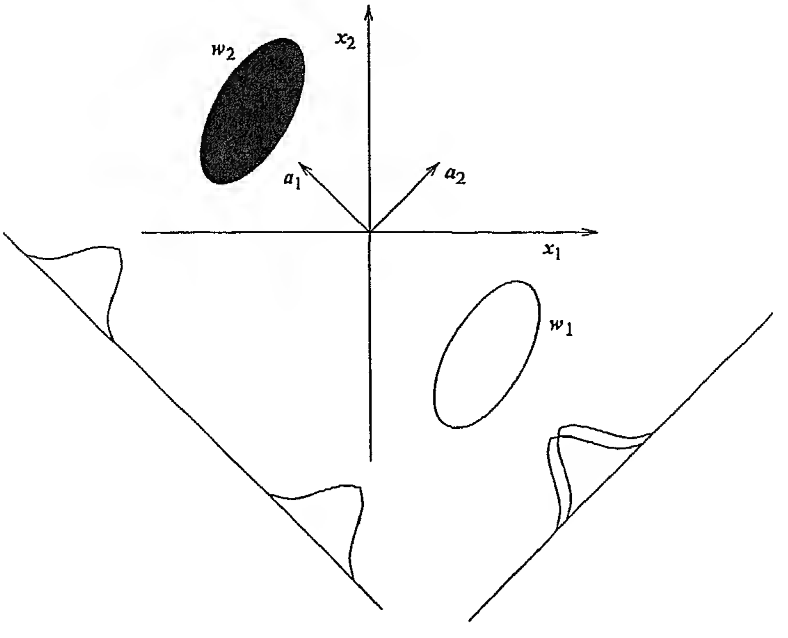


Figure 3 4 PCA applied to two dimensional two class problem In this example one eigenvector separates the two classes well and another eigenvector doesn t So the only one eigenvector components which provide high separability of the classes are enough for classification problem For reference of this figure see [14]

zero mean probability distribution $p(\mathbf{x})$ Let $\mathbf{u}$ denote a vector in $\mathbf{R}^n$ onto which the input vectors are to be projected The projection $\mathbf{x}^T\mathbf{u}$ is the linear sum of n zero mean random variables which is itself a zero-mean random variable Here the objective is to find the solution(s) $\mathbf{u}^*$ that maximises $\langle(\mathbf{x}^T\mathbf{u})^2\rangle$ the variance of the projection $\mathbf{x}^T\mathbf{u}$ with respect to $p(\mathbf{x})$, subject to $||\mathbf{u}|| = 1$ In other words we are interested in finding the maxima $\mathbf{w}^*$ of the criterion function

$$J(\mathbf{w}) = \left\langle \left(\mathbf{x}^T \frac{\mathbf{w}}{||\mathbf{w}||}\right)^2 \right\rangle \qquad (3\ 2)$$

from which the unity norm solution(s) $\mathbf{u}$ can be computed as $\mathbf{u}^* = \mathbf{w}^*/||\mathbf{w}^*||$, with $||\mathbf{w}^*|| \neq 0$ Now by noting that

$$\left\langle (\mathbf{x}^T\mathbf{w})^2 \right\rangle = \langle (\mathbf{x}^T\mathbf{w})\ (\mathbf{x}^T\mathbf{w}) \rangle = \mathbf{w}^T \langle \mathbf{x}\mathbf{x}^T \rangle\ \mathbf{w}$$

and recalling that $\langle \mathrm{xx^T} \rangle$ is the autocorrelation matrix C Equation 3 2 may be expressed as

$$J(\mathrm{w}) = \frac{\mathrm{w^T C w}}{||\mathrm{w}||^2} \tag{3 3}$$

The extreme points of $J(w)$ are the solutions to $\Delta J(\mathrm{w}) = 0$, which gives

$$\begin{aligned} \Delta J(\mathrm{w}) &= \frac{\mathrm{Cw}||\mathrm{w}||^2 - (\mathrm{w^T C w})\mathrm{w}}{||\mathrm{w}||^4} = 0 \\ or \quad \mathrm{Cw} &= \frac{(\mathrm{w^T C w})}{||\mathrm{w}||^2}\mathrm{w} \end{aligned} \tag{3 4}$$

The solutions to Equation 3 4 are $\mathrm{w} = a\mathbf{c}^i$, $i = 1, 2, \quad , n$, $a \in \mathbf{R}$ In other words, the maxima w of $J(\mathrm{w})$ must point in the same or opposite direction as one of the eigenvectors of C For the maximum exists at $\mathrm{w} \; = a\mathrm{c}^1$ for some finite real valued a Therefore, the variance of the projection $\mathbf{x^T u}$ is maximized for $\mathbf{u} = \mathbf{u}_1 = \mathrm{w}^*/||\mathrm{w} \; || = \pm \mathbf{c}^1$ Next we repeat the preceding maximization of $J(\mathrm{w})$ in Equation 3 3 but with the additional requirement that the vector $\mathbf{w}$ be orthogonal to $\mathbf{c}^1$ Let it is $\mathbf{w}^* = a\mathbf{c}^2$ Thus $\mathrm{u}_2 = \mathrm{c}^2$ Similarly the solution $\mathrm{u}_3 = \mathrm{c}^3$ maximizes J under the constraint that $\mathbf{u}_3$ be orthogonal to $\mathbf{u}_1$ and $\mathbf{u}_2$ simultaneously Continuing this way, we arrive the solution at the m principal directions $\mathbf{u}_1$ through $\mathbf{u}_m$ The projections $\mathbf{x^T u}_i$, $i = 1, 2, \quad , m$, are called the *principal components* of the data For reference of PCA see [6]

Result 4 features

We can use PCA technique to do linear transformation from input space to reduced dimensional space and also to reduce the dimensions by finding the number of redundant features It is useful in our problem because we want to know the minimum number of features required for the classification with good enough accuracy We also wanted to know the features which are really not affecting the classifiers performance

For that first we found the eigenvalues in all 6 principal components eigenvector directions using PCA with large data set We compared all 6 eigenvalues and found that out of 6 eigenvalues, 4 are comparable and two are very less compared to other 4 So, we decided that there are two features which are really redundant

We tested the classifiers with selecting one feature as a redundant and tested with only rest five features and the results were compared We also calculate the eigenvalues after removing one feature and compared with those before removing one feature If it is same then the removed feature is redundant otherwise not While doing testing with all features considering as a redundant but one at a time and while comparing the results it is found that duration is redundant Then the

similar test is done for searching another redundant feature It is found that rise time is also second redundant It is because impulsive transient and oscillatory transients can be classified using affective values of peak value and dominant interharmonics in case of oscillatory transients So the duration and rise time are really redundant here The rest four features e g peak voltage, RMS voltage THD and Dominant Interharmonics are the optimum for these six disturbances set as shown as first 6 in Table 3 3 One can include no disturbance as the seventh disturbance class because it doesn t need any extra feature except these four optimum features mentioned above The no disturbance is also included in the same table Our final generalisation testing of different classifiers will be done using these four features only for these seven disturbance classes

4 Analysed Classifiers

Before analysis of all classifiers, we explain here the basic structure of the classifier construction and functioning processes The simple structure for constructing the classifier in general is shown in Figure 4 1 *Training set* is a set of data samples with the information about their belongingness which is used to design the classifier It is also known as *Knowledge dataset* or *Learning set* During construction of the classifier data samples of the same class are used to create the *functional parameters* or *discriminant set* of that class It is shown as M_i for i^{th} class in figure mentioned above g defines the function to construct M_i using training subset T_i Figure 4 2 shows general structure for classifier operation to classify an unknown input vector x The *Decision function* f uses the discriminant set of every class from designed classifier to calculate the belongingness of input vector x in a particular class in a specific form The *class decider* decides the output class from all class function values

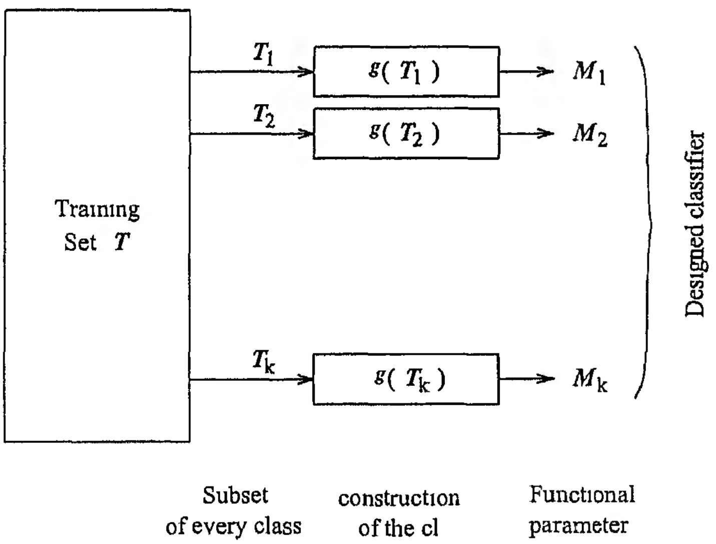


Figure 4 1 Basic structure of classifier construction

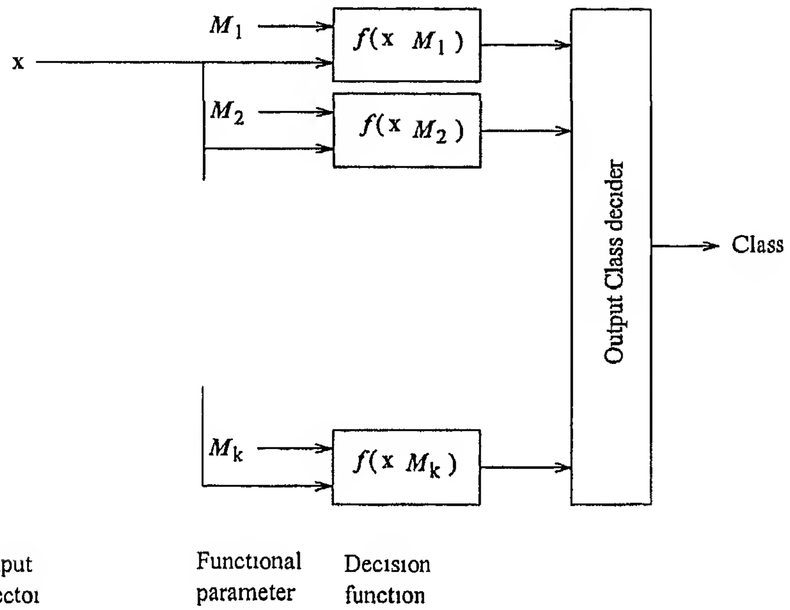


Figure 4 2 Basic structure of classifier operation

4 1 Euclidean distance classifiers

It is one of the simplest pattern classifiers It is also called as nearest mean linear classifier or minimum distance classifiers Each object is assigned to the class for which the mean is nearest (in the Euclidean sense) in feature space This classifier implements a piece wise linear discrimination function between these means Here, the decision boundaries are the points that are equally distant from two or more of the class templates With an Euclidean distance method the decision boundary between region i and region j is the line or plane that is the perpendicular bisector of the line from class templates μ_i to μ_j Analytically these linear boundaries are a consequence of the fact that the discriminant functions are linear

For 2 dimensional feature space, it i straight line or a set of lines For 3 dimensions, a plane or set of planes and for n dimensions $n > 3$ it is a hyperplane or set of hyperplanes How well the classifier works depends upon how closely the input patterns to be classified resembles the templates Two dimensional patterns with Gaussian distribution, classified by Euclidean distance classifier is shown in figure 4 3 Here, the x_1 and x_2 are two features of the pattern vectors From the figure, one can observe that the Euclidean distance classifier misclassifies the data samples if they lie on the other side of the decision bound

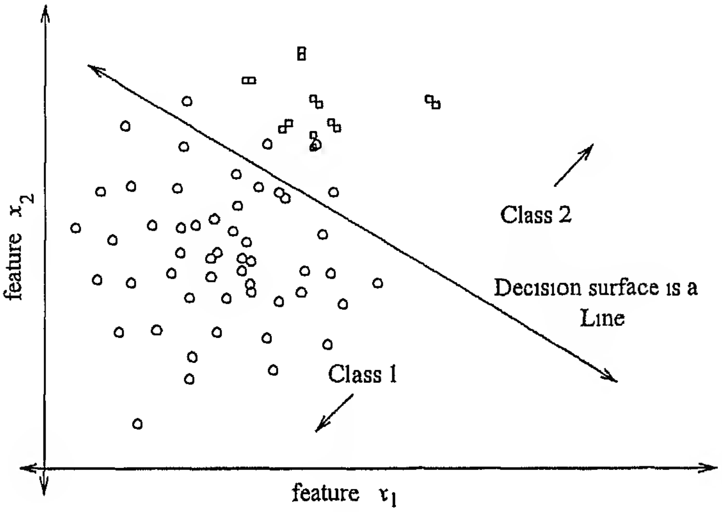


Figure 4 3 Euclidean distance classifier with normally distributed patterns

ary from it's own class side The performance of the classifier is dependent not only on the data distribution of individual class, but comparative distribution of the data of neighbour classes It is also affected by large variation in the size of regions of data points of neighbour classes There are some limitations to this this classifier

The block diagram of Euclidean distance classifier as shown in Figure 4 4 explains about the design of classifier and classification of the unknown input vector Knowledge database is provided for design of the classifier Classifier calculates the mean vector for every class as a functional parameter This is enough for the requirements to classify the unknown data sample The decision function is Euclidean distance of the input vector to all classes' mean vectors and the final class of belongingness will decided by final discriminating min function on Euclidean distances from all the classes' mean

Template matching can easily be expressed mathematically Let x be the feature vector for the unknown input, and let $\mu_1, \mu_2, \quad , \mu_k$ be templates for k classes Then the Euclidean distance or the error in matching $\mathbf{x}$ against μ_q is given by Equation 4 1 A minimum error classifier computes d_q for $q = 1, \quad , k$ and chooses the class for which this error is minimum Since, d_i is also the distance from x to μ_q it is called as a minimum distance classifier The distance is Euclidean (linear), so we call this a *Euclidean distance* classifier[2] The mathematical representation of class assignment to input vector x is stated in Equation

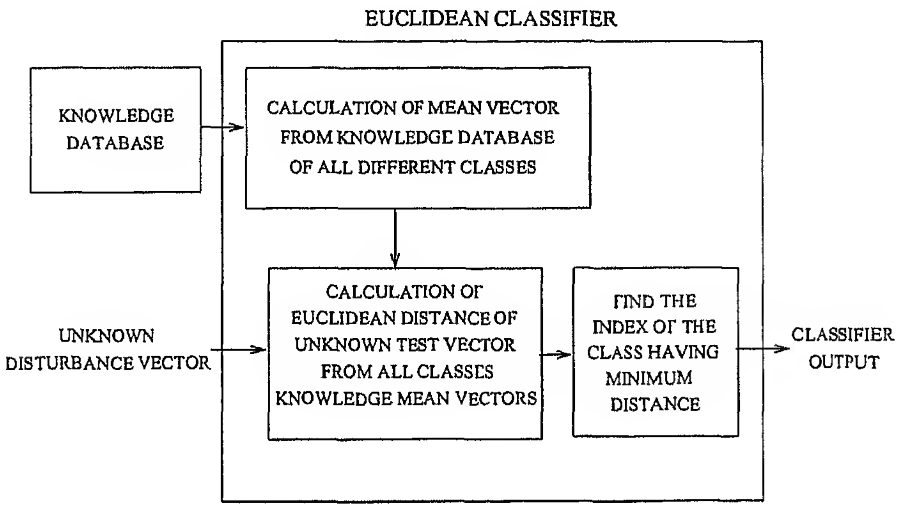


Figure 4 4 Euclidean minimum distance classifier block diagram

4 2

$$
\begin{aligned}
d_q &= \sqrt{(\mathrm{x}-\mu_q)^T\,[\mathrm{I}]\,(\mathrm{x}-\mu_q)} = \sqrt{\sum_{i=1}^{n}(x_i - m_{q\,i})^2} \\
\text{Where}\quad \mathrm{x} &= [x_1, x_2, \quad , x_n]^T \in \mathrm{R}^n, \\
\mathrm{I} &= \text{unity matrix of size } (n \times n) \\
\text{and}\quad \mu_q &= mean_{i=1}^{N_q}\, \mathrm{p}^q \\
&= [m_{q\,1}, m_{q\,2}, \quad , m_{q\,n}] \in \mathrm{R}^n \\
\text{Where}\quad N_q &= \text{number of data points of } q^{th} \text{ in training set} \\
\text{and}\quad \mathrm{p}_i^q &= i^{th} \text{ vector of } q^{th} \text{ class from training set} && (4\ 1) \\
S(\mathrm{x}) &= arg\, min_{q=1}^{k}\, \|\mathrm{x}-\mathrm{m}_q\| && (4\ 2)
\end{aligned}
$$

Because of the it's basic principle is linear distance, it has certain limitations to the real applications It is required to modify the data in input space before using for the classifier to have a good performance

4 2 Bayes classifier

The Bayes classifier is a mechanism which minimises the classification error This one is a probabilistic classifier It works on the basic principle of Gaussian prob ability density functions (PDF) of the data points in all classes In figure 4 5, the

Bayes classifier follows Caussian probability functions is shown for two classes with only one feature The classification strategy here is labeling an object with the label foi which the Bayes probability is highest The data distiibutions of both the classes are assumed to be normal (Gaussian) So their PDFs will be same as shown in figure The hatched area represents the error this function $S(x)$ makes Theoretically the minimum eiror this classifier makes called the Bayes error

This is a statistical model with assumed PDFs Gaussian (normal) distribu

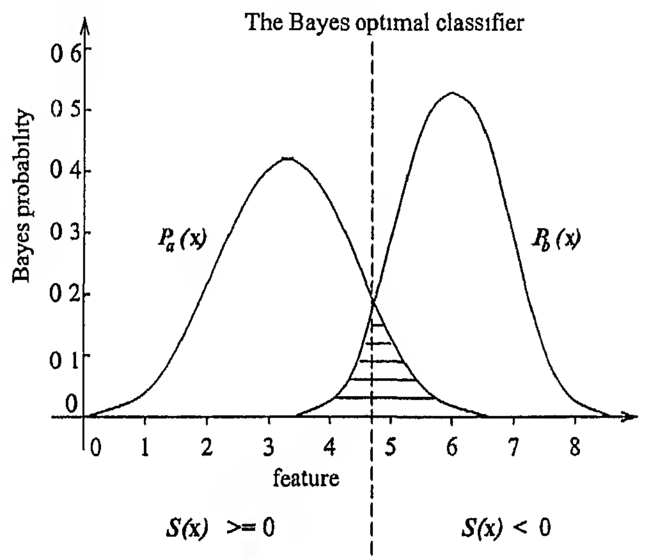


Figure 4 5 Bayes classifier with Gaussian PDF function in one dimensional feature space For reference see [12]

tions are usually used for this purpose Block diagram of the classifier is shown in Figuie 4 6 The parameters of each distribution mean vector and covariance matiix are estimated on the basis of the training samples of each class Next, the estimated PDFs of all classes are combined in order to classify unknown samples We used here Bayesian Quadratic classifier strategy Each PDΓ is calculated for an unknown sample, the the sample is classified to a class with a highest value Its classification boundary forms a quadratic curve (or a quadratic surface in a higher dimensional space)

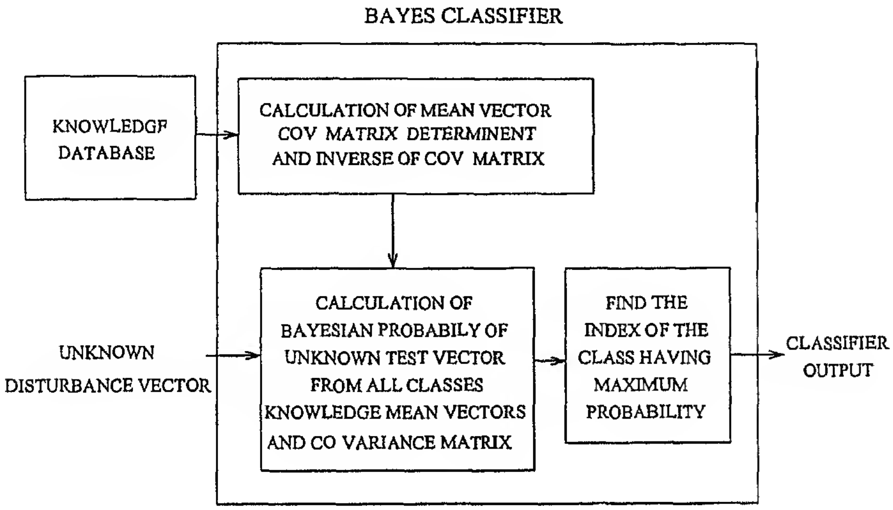


Figure 4 6 Bayes classifier block diagram

4 2 1 Bayes Decision Theory

The Bayes classifier is based on the assumption that all of the relevant probability values are known The *a priori* probability $P(w_i)$ for $i = 1, \; , k$ are assumed to be known If no information is available of *a priori* probability then one can consider unity for all classes Then it is called *maximum likelyhood classifier* So *a priori* probability factor is ignored here The random variable x can be determined to what class it belongs to based on a decision rule of probabilities[9]

$$\text{Decide } w_i \text{ for } maxP(w_i|\mathrm{x}) \text{ for } i = 1, 2, \; , k \tag{4 3}$$

The *a posteriori* probabilities $P(w_i|\mathrm{x})$ may be calculated from *a priori* probabilities $P(w_i)$ and the conditional density functions $p(\mathrm{x}|w_i)$ using Bayes' theorem, which is

$$\begin{aligned} P(w_i|\mathrm{x}) &= \frac{p(\mathrm{x}|w_i)P(w_i)}{p(\mathrm{x})} \\ \text{where } p(\mathrm{x}) &= \sum_{j=1}^{k} p(\mathrm{x}|w_i)P(w_i) \end{aligned} \tag{4 4}$$

The representation of the classifier is in terms of a set of discriminant functions $g_i(\mathrm{x}), i = 1, \; , k$ where c is the number of classes The classifier is said to assign a feature vector x to class w_i if

$$g_i(\mathrm{x}) > g_j(\mathrm{x}) \text{ for all } j \neq i \tag{4 5}$$

For minimum error rate classification the discriminant function can be expressed as

$$g_i(\mathrm{x}) = P(w_i|\mathrm{x}) \tag{4.6}$$

The effect of any decision rule is to divide the feature space into c decision regions $R_1, \ldots, R_k$. If $g_i(\mathrm{x}) > g_j(\mathrm{x})$, for all $j \neq i$, then x in R_i and the decision rule calls for us to assign x to w_i. The Bayes classifier assumes a probability distribution is known for the observation. In this case the distribution chosen was the Gaussian or normal distribution. The general multivariate density for this distribution is expressed as

$$p(\mathrm{x}) = \frac{1}{(2\pi)^{\frac{d}{2}}|\mathrm{K}|^{\frac{1}{2}}} exp[-\frac{1}{2}(\mathrm{x}-\mu)^T\mathrm{K}^{-1}(\mathrm{x}-\mu)] \tag{4.7}$$

where x is a n component column vector. μ is a n component *mean vector*. K is the n by n *covariance matrix*, K^{-1} is the inverse of K, and $|\mathrm{K}|$ is the determinant of K. Also $\mu = E[\mathrm{x}]$ and $\mathrm{K} = E[(\mathrm{x}-\mu)(\mathrm{x}-\mu)^t]$. The covariance matrix K is always symmetric and positive semidefinite. The diagonal elements of K is the variance and off diagonal elements are the covariances. For the Bayes probability of test vectors x belongs to i^{th} class is considered as defined in Equation 4.7 with the covariance matrix K is consider for i^{th} class, i.e. K_i. If no co-variance exist between different features then $\mathrm{K} = \mathrm{I}$ and the Bayes classifiers works like Euclidean classifier. The mathematical representation of class assignment to input vector x is as follows

$$S(\mathrm{x}) = arg\,max_{i=1}^{k}\, p_i(\mathrm{x}) \tag{4.8}$$

In practical problem however, an infinite number of learning objects is never available. For that reason, the true probability density functions and a priori probability information is not available. In this case the error will probably be higher than Bayes error. Classifier construction is then based on the assumption that the learning objects available represent the true probability density functions. It is known as an ideal classifier because it has a highest generalisation performance over other classifiers. The other classifiers are compared to this classifiers to evaluate their performances. The limitation of this classifier is data distribution of the knowledge data.

4.3 Euclidean k-nearest neighbour classifier

Nearest neighbour algorithms are a famous, attractive and simple form of non parametric classifier. Given a set of stored training samples (prototypes), an unknown sample x_o will be classified by computing a distance in input space from x_o to each of the training samples. It is assigned to the class label Ω which

is a label of the nearest neighbour A neighbour is deemed nearest if it has the smallest distance in the Euclidean sense in feature space Euclidean distance is the simplest of all type of distances
NN classifier is a special case of k NN classifier where k=1 The k nearest proto

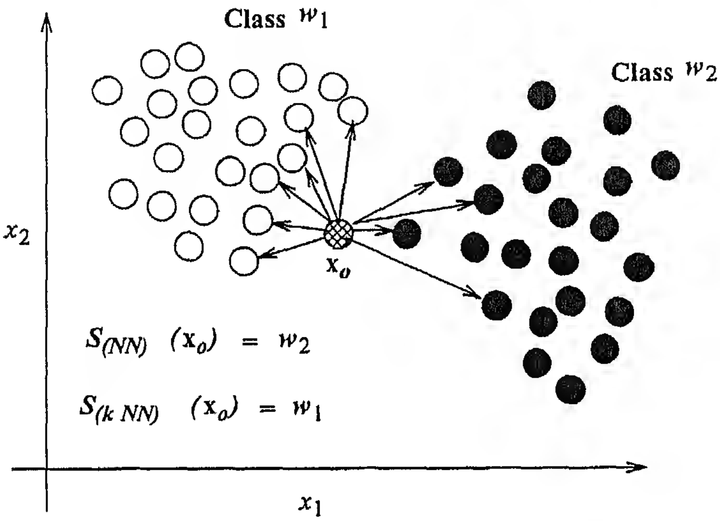


Figure 4 7 Basic principle of k nearest neighbour classifier

types will be selected to calculate the class of the unknown sample x_o by majority decision of the k nearest neighbours In another words, k NNC selects those locates whose rank distances from input x_o are less than or equal to k Then the decision of input x_o will be more frequently occurring class among chosen k nearest neighbours The value of k, number of neighbours to be considered is chosen by the user It is normally 1 5% of the average data points of one class in the input space The Basic principle of k nearest neighbour classifier is shown in figure 4 7 for k = 10
Block diagram of k nearest neighbour classifier developed here is shown in Figure 4 8 All knowledge data samples are used to calculate the Euclidean distance from the unknown input vector in k NNC as stated in block diagram To classify the vector, two classifiers works in parallel First is NNC which classifies the input vector by searching the nearest neighbour and labeling it s class belongingness to unknown vector as mentioned before The second one is k NNC, which provides the information about the quality of classification It finds the classes of majority belongingness as mentioned previously from k NN and calculates the number of neighbours of every class belongingness It provides the information about the location of input vector in the space and and it s quality by having idea about the percent belongingness to other classes

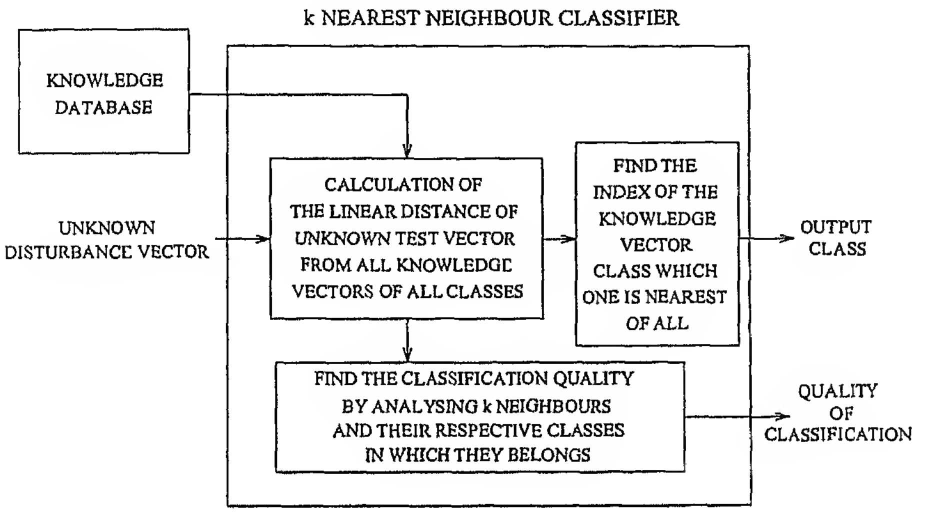


Figure 4.8 k nearest neighbour classifier block diagram

Mathematically, NN classifier can be represented as defined in Equation 4.9 and k NNC can be represented as defined in Equation 4.10

$$
\begin{aligned}
S(\mathrm{x}_o) &= S(\mathrm{x}^j) \\
\text{where } j &= arg\ \ min_{p=1}^{N_{all}} ||\mathrm{x}_o - \mathrm{x}^p|| \\
\text{where } N_{all} &= \sum_{i=1}^{k} N_i && (4.9) \\
\Omega &= S(\mathrm{x}_o) = arg\, max_{i=1}^{k}\, f_i \\
\text{where } \ f_i &= \text{number of neighbours from } i^{th} \text{ category} && (4.10)
\end{aligned}
$$

This discrimination function or decision surface by NN classifier will in general be jagged, piece wise linear function since it is influenced by each object available in the learning set. It becomes smoother in k NNC because the decision is judged by k points rather than only one point as in NNC. A disadvantage of this method is its large computing power and storage requirements since for classifying an object its distance to all the objects in the learning set has to be calculated.

4.4 Fuzzy classifier

The fuzzy rule based classification approach is sufficiently transparent so that designer can understand the decision process and easily apply changes and/or modifications to it. There are different ways to develop fuzzy rules and fuzzy

membership functions from training dataset Every class has one fuzzy rule and every fuzzy rule has one membership function from every feature Membership functions are parameterised using min max limits clustering of training samples of the same class and projecting every cluster on all features For example creation of e for three classes in two dimensional us method only average of all data points and width of cluster over all features is considered to create limit parameters of membership functions on every feature For reference to this classifier please refer [10]

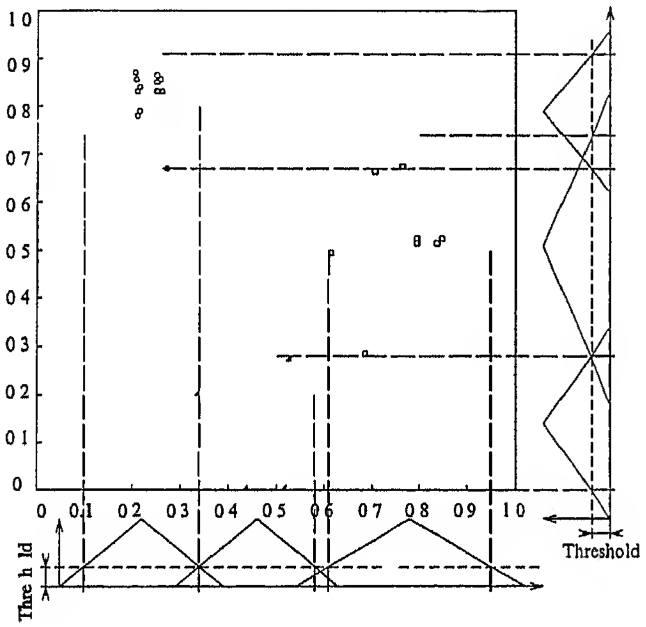


Figure 4 9 Creating equilateral triangular membership functions through performing projections of min max limits onto the individual feature axes

4 4 1 Assumptions for design of fuzzy classifier

- The data distribution over all features is continuous
- Distribution of the data points inside the cluster is symmetrical about the mid point of the region of cluster
- Intersection of two different class clusters is empty

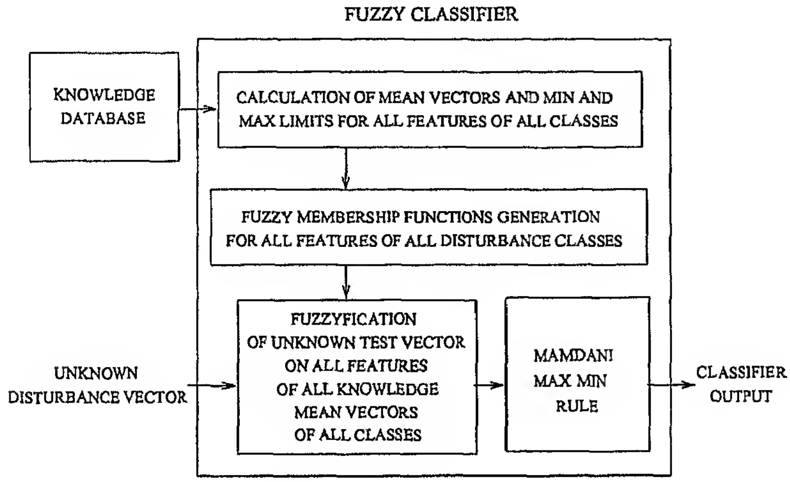


Figure 4.10 Fuzzy pure classifier block diagram

Block diagram of fuzzy classifier developed here is shown in Figure 4.10. The requirement of the fuzzy classifier are mean values and minimum and maximum limits for every feature range of all classes. It will generate the membership function for every feature of all classes with that information. If the knowledge data set is provided then it will extract this information from this data set. It can calculate the membership function values for all feature components of the input vector $\mathbf{x}$ and finally the output class will be evaluated using max min Mamdani rule on membership value matrix of size $(n \times k)$.

4.4.2 Creation of membership functions

The patterns are vectors $\mathbf{x} = [x_1, \ldots, x_n] \in \mathbf{R}^n$ and set of k classes is a crisp subset of $\mathbf{R}$. The pattern features are represented by fuzzy sets and the classification is described by a set of linguistic rules. There are variable number of fuzzy sets for different features. Let p_{ij} be representation of j^{th} feature of i^{th} class pattern $\mathbf{p}_i$ from training set. p_j be representation for a sample value of j^{th} feature. Let μ_{ij} be a membership function of i^{th} class for j^{th} feature

Mathematically the steps of membership function generation are as follows

$$
\begin{aligned}
p_j^{min} &= min_{q=1}^{N}\, p_{ij} && (4\ 11)\\
p_j^{max} &= max_{q=1}^{N}\, p_{j} && (4\ 12)\\
p_j^{av} &= mean_{q=1}^{N}\, p_{j} && (4\ 13)\\
p_j^{width} &= p_j^{max} - p_j^{min} && (4\ 14)\\
p_j^{1} &= p_{ij}^{max} - p_{ij}^{a} && (4\ 15)\\
p_{ij}^{2} &= p_{ij}^{ave} - p_{ij}^{min} && (4\ 16)\\
p^{minwidth} &= min_{j=1}^{n}\,(min_{i=1}^{k}\, p_{j}) && (4\ 17)\\
p^{maxwidth} &= max_{j=1}^{n}\,(max_{i=1}^{k}\, p_{j}) && (4\ 18)\\
factor &= \frac{p^{minwidth}}{p^{maxwidth}} && (4\ 19)\\
p_j^{newwidth} &= p_j^{width}\,(1 + factor) && (4\ 20)\\
\mu_{ij}(x_j) &= max\,(min\,(l_{ij}^{1}, l_{ij}^{2}), 0) &&\\
\text{where } l_j^{1} &= \frac{1}{max(p_{ij}^{1}, p_j^{2})}(x_j - p_j^{ave}) + 1 &&\\
\text{and } l_j^{2} &= \frac{1}{max(p_j^{1}, p_j^{2})}(-x_j + p_j^{a}) + 1 && (4\ 21)
\end{aligned}
$$

4 4 3 Fuzzy classification strategy

It is a Mamdani fuzzy inference rule called Max min fuzzy rule we implemented for fuzzy classification The $max - min$ concept used here for $t - norms$ and $t - co - norms$ is stated below

$$
\begin{aligned}
&\textit{fuzzy}\,\text{AND} && \mu(\mathrm{x}) = min[\mu_1(x_1), \ \ , \mu_n(x_n)] &&\\
&\textit{fuzzy}\,\text{OR} && \mu(\mathrm{x}) = max[\mu_1(x_1), \ \ , \mu_n(x_n)] &&\\
&\textit{fuzzy}\,\text{NOT} && \mu_j(\mathrm{x}) = 1 - \mu_j(x_j) && (4\ 22)
\end{aligned}
$$

The *Linguistic Form* of fuzzy min rule is as follows

$$\text{IF}\ x_1 = A_1\ \text{AND}\ x_2 = B_1\ \text{THEN}\ \mathrm{x} = C_1 \qquad (4\ 23)$$

For example, Mamdani fuzzy inference rule for two classes and two features is shown in Figure 4 11 Here two features defined as **X** and **Y** and two fuzzy **AND** rules are there A_1, B_1 and A_2, B_2 are membership functions of features **X**, **Y** for rule 1 and rule 2 respectively As shown in figure, each rule uses min function for each output calculation Output is Z The max function can be applied for final defuzzified value of Z The defuzzification technique for output

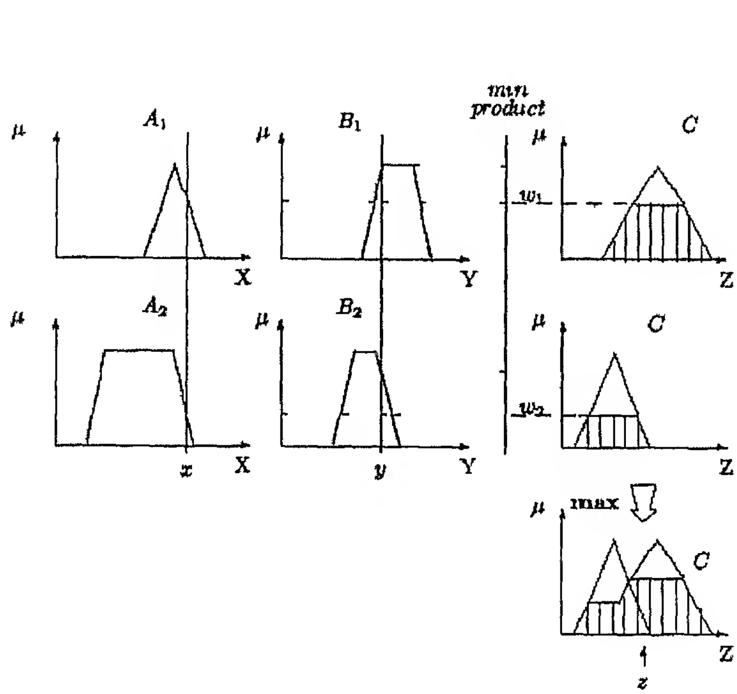


Figure 4 11 Mamdani fuzzy model for two features and two classes

from Mamdani rule can be Center of area method as shown in Equation 4 24 Instead of doing Defuzzification using ***centre of area*** method, max rule is applied for final class identification of the class The $max - min$ Mamdani rule which here applied is stated in Equation 4 25

$$z_{out} = \frac{\int_z \mu_c(z) z dz}{\int_z \mu_c(z) dz} \tag{4 24}$$

$$S(\mathrm{x}) = arg\, max^k_{=1} min^n_{j=1} \mu_{ij}(x_j) \tag{4 25}$$

4 5 RCE neural network classifier

Restricted Coulomb Energy (RCE) classifier works on the principle of hyper spherical classifiers having hyperspherical decision boundaries RCE classifier is a potential function governing mapping characteristics interpreted as a restricted form of a *high dimensional Coulomb potential* between a positive test charge and negative charges placed at various sites[7] The RCE net is capable of developing proper separating boundaries for nonlinearly separable problems Block diagram of RCE classifier developed here is shown in Figure 4 12 The previously trained RCE net is used for classification of unknown samples Every input vector will

be provided by output binary vector after analysis by the trained network The length of the output binary vector will be same as the number of classes The index of the binary vector represents the class So output class index value of this output vector is assigned logic 1 and all other class index values are assigned logic 0 The class decider detects the output class from this output vector

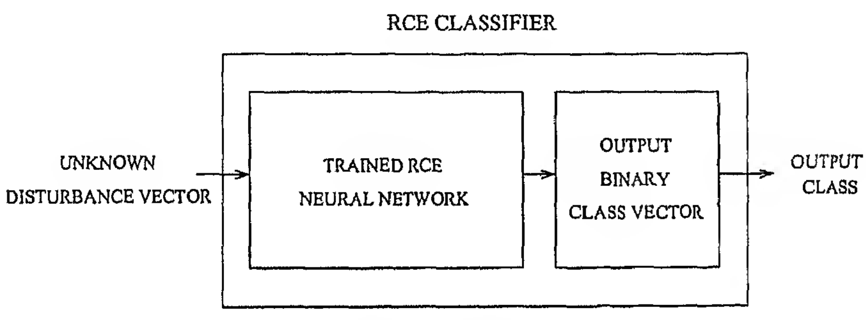


Figure 4 12 RCE classifier block diagram

4 5 1 Basic principle of hyperspherical classifiers

It stores the example patterns in Euclidean space like in nearest neighbour classifier and calculates the linear distance between the new point and the the known points in the input space Each stored point has a finite *radius* that defines its region of influence Interior of the hypersphere generated represents the decision region associated with center point's category The finite radii of the regions of influence can make a hyperspherical classifier abstain from classifying patterns from unknown categories Thus this later feature enhances the classifier's ability to reject *rubbish*

4 5 2 Classifier Development

RCE classifier is a mapping from real to binary, defined as follows

$$O \quad \mathrm{x} \in \mathbf{R}^n \Rightarrow \{0, 1\}^k \tag{4 26}$$

$$\text{So,} \quad \mathrm{O}_q(\mathrm{x}) = \begin{cases} 1 & \text{if x belongs to } q^{th} \text{ class} \\ 0 & \text{if x doesn't belong to } q^{th} \text{ class} \end{cases} \tag{4 27}$$

$$S_q(\mathrm{x}) = \omega_q, \quad if(O)_q = 1 \tag{4 28}$$

The architecture of the RCE network contains two layers a hidden layer and an output layer The hidden layer is fully interconnected to all components of an input vector $\mathrm{x} \in \mathbf{R}^n$ The output layer consists of L units The output

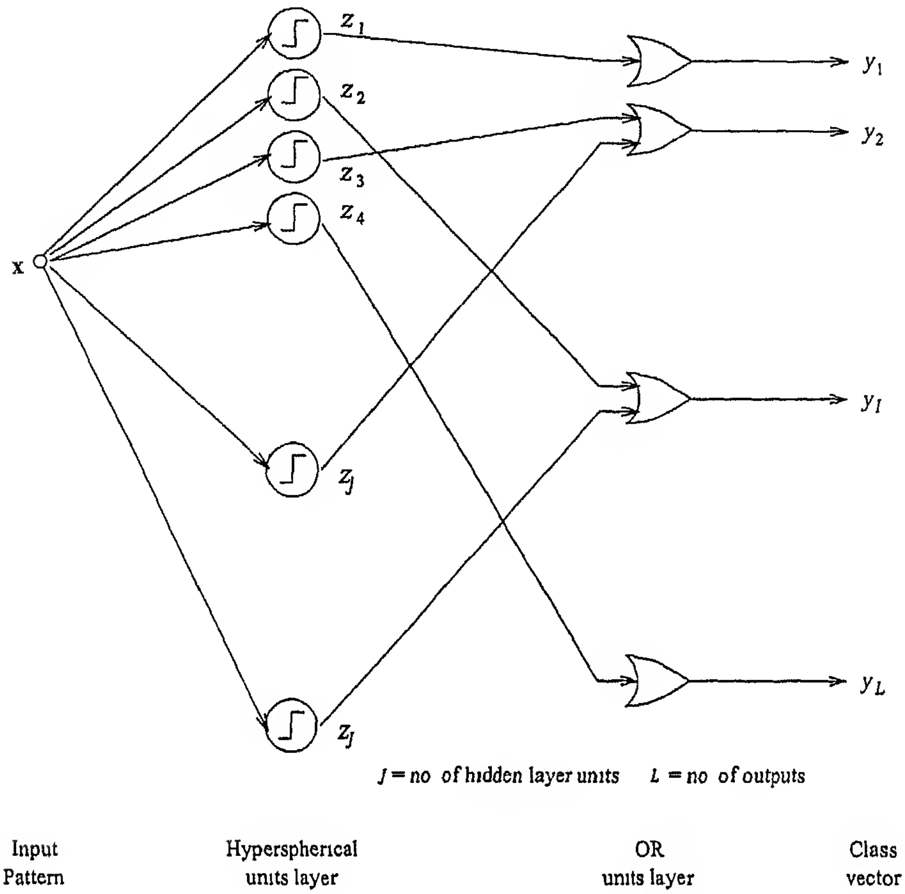


Figure 4.13 RCE network architecture

layer is sparsely connected to hidden layer, each hidden unit projects its output to one and only one output unit. The architecture of RCE net is shown in Figure 4.13. Each unit in the output layer corresponds to a pattern category. The network assigns an input pattern to a category l if the output cell y_l is activated in response to the input. The decision of the network is *unambiguous* if one and only one output unit is active upon the representation of the input, otherwise the decision is said to be *ambiguous*. The transfer characteristics of the j^{th} hidden unit is given by

$$z_j(\mathbf{x}) = f[r_j - D(\mu_j, \mathbf{x})] \tag{4.29}$$

where $\mu_j \in \mathbf{R}^n$ is a parameter vector called center, $r_j \in \mathbf{R}$ is a threshold or *radius*, and D is linear distance function between two vectors. Here, f is the

threshold activation function given by

$$f(\xi) = \begin{cases} 1 & \text{if } \xi \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{4 30}$$

On the other hand, the transfer function of a unit in the output layer is the logical **OR** function The j^{th} hidden unit in the RCE net is associated with a hyperspherical region of the input space that defines the unit s region of influence The location of this region is defined by the center μ_j and its size is determined by the radius r_j According to equation (4 29) any input pattern falling within the influence region of a hidden unit will cause this unit fire This hidden units define a collection of hyperspheres in the space of input patterns Some of these hyperspheres may overlap When a pattern falls within the region of influence of several hidden units, they will all *fire* and switch on the output units they are connected to

4 5 3 Training of RCE net

Training of RCE net involves two mechanisms unit commitment and modifica tion of hidden unit radii[7] unit commitment involves hidden layer units and output layer units

Initially the network starts with no units An arbitrary sample pattern $\mathbf{x}^1$ is selected from the training set, and one hidden unit and one output unit are al located The allocated hidden unit center μ_1 is set equal to $\mathbf{x}^1$ and its radius r_1 is set equal to a user defined parameter r_{max} (r_{max} is the maximum size of the region of influence of a hidden unit) This unit is made fully interconnected to the input pattern and projects its output z_1 to the allocated output unit (OR gate) This output unit represents the category of the input $\mathbf{x}^1$ Next a second arbitrary example $\mathbf{x}^2$ is chosen and fed into the current network Here one of three scenarios emerges

First if $\mathbf{x}^2$ causes *output unit to fire, and* $\mathbf{x}^2$ *belongs to the category represented by this unit,* then nothing is done and the training is continued with a new input If this same scenario occurs during training when network has multiple hidden and output units representing various categories if the only correct output unit fires then nothing is done But if correct unit may fire with one or more output units of other categories then the radii of the hidden active units representing other categories are reduced until they become inactive

Second, $\mathbf{x}^2$ *doesn't cause to fire output unit even if it belongs to that category* then a new hidden unit is allocated with center at $\mu_2 = \mathbf{x}^2$ and radius r_{max} and z_2 is connected to the correct output unit In general radius of new center is $r = min(r_{max}, d_{min})$, where d_{min} is the distance from this new center to the nearest center of a hidden unit representing any other category This setting of radius may cause one more output units to fire along with correct category reduction of radii of active hidden units of other categories is done like in the

first scenario

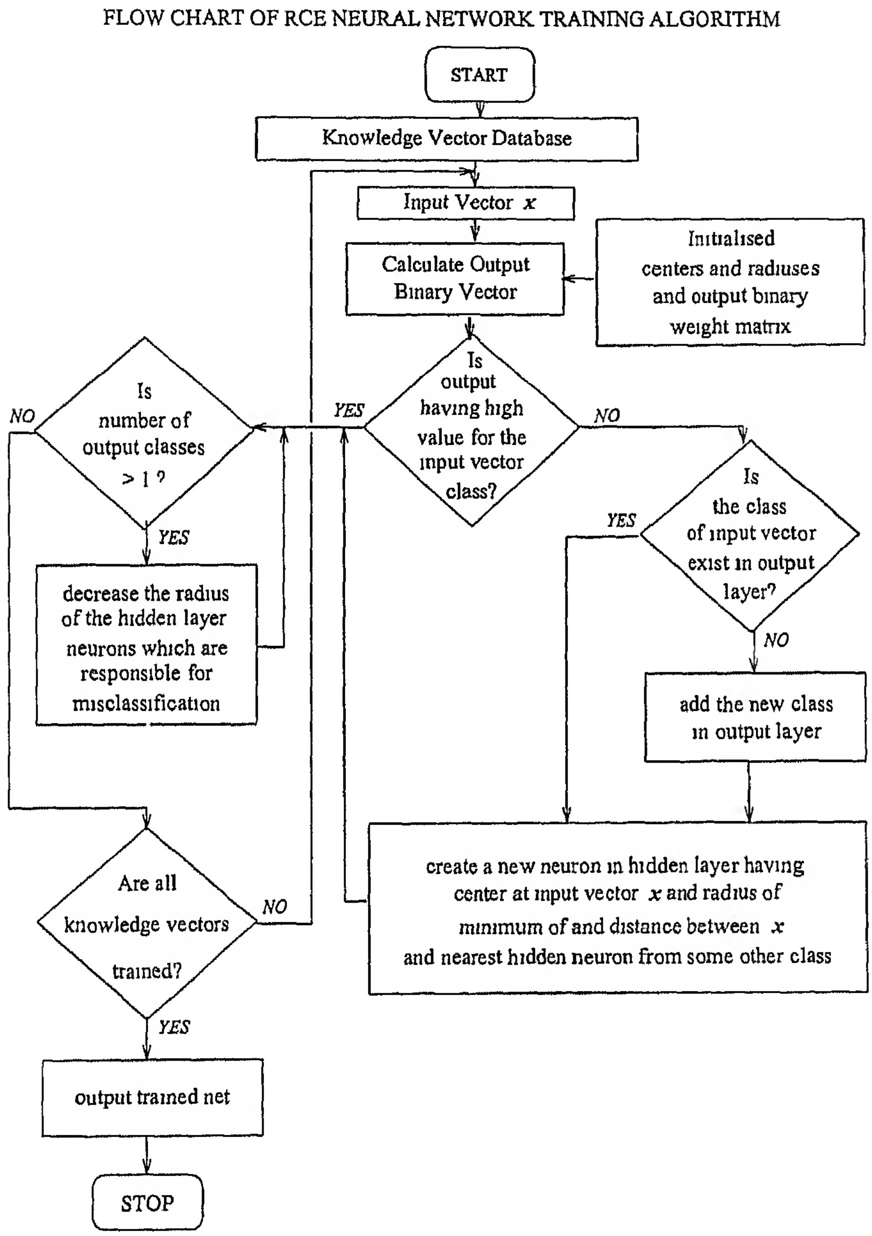


Figure 4 14 RCE net training flow chart

Third and Final *input belongs to new category that is not represented by the*

network Here as in the first step of training procedure, a hidden unit centered at this input is allocated and its radius is set as in second scenario Also a new output unit representing the new category is added that receives an input from a newly allocated hidden unit Again if existing hidden units become active under this scenario then their radii are shrunk until they become inactive The training phase continue until all set of examples are finished and no new units are allocated and the size of the regions of influence of all hidden units converges The flow chart for RCE net training is shown in figure 4 14

4 6 Sequential Fuzzy Classifier

The features and techniques of a special structure of fuzzy classifier that was developed for the purpose of designing a mix disturbance classification system are described here

To give an overview one can identify the method as sequential set of fuzzy rules Number of features considered before reaching a decision may vary at different parts of the classification tree The aim of this system is to judge the type of input unknown disturbance if it is a mixture of two or more known disturbances The classifier is designed in such a way that if the input vector is having characteristics of two different classes then in that case, sequential classifier produces results of all possible classes of belongingness Here, the selections of features which allows user to define constraints of the classifier to classify different type of mixed disturbances Here the classification of mixed disturbances is feature dependent The feature for mix disturbances is selected along with the disturbances dependent on that feature on only So, every feature having mixing nature is responsible to judge one disturbance if it is when the classification tree is executed during sequential classification So, if there is at least one disturbance except *no disturbance* depends on only one feature then it is possible to use this classification technique For reference to this classifier, please refer [4] and [5]

The block diagram of Fuzzy Sequential classifier is shown in Figure 4 15 It's requirements are almost same as fuzzy pure classifier as studied in Section 4 4 It need additional information about the features and their related disturbances for establishing the fuzzy mixed classifier We will discuss it in the next section in detail Outputs from both fuzzy pure classifier and fuzzy mixed classifier are used as a total output Normally fuzzy pure classifier is the first part of fuzzy sequential classifier and second part is fuzzy mixed classifier Here, we will discuss fuzzy mixed classifier in detail since fuzzy pure classifier we already studied

4 6 1 Description

The knowledge database consists of training set of disturbances including normal voltage with no disturbances The generation of fuzzy sets for all disturbances

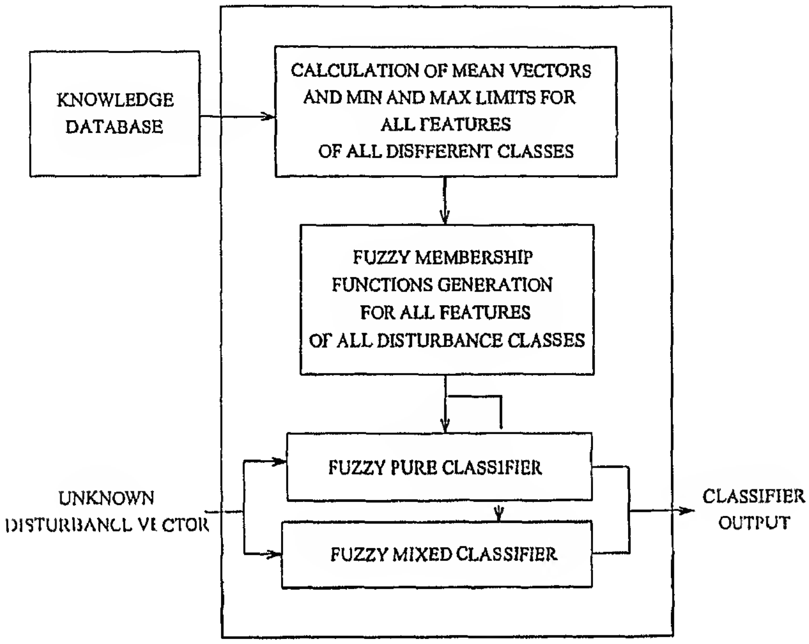


Figure 4 15 Block diagram fuzzy sequential classifier

on all features are done using training dataset as same as in normal fuzzy pure classifier mentioned before The equilateral triangle membership function could be used for all classes The test vector is first tested with fuzzy pure classifier and fuzzy sequential or mix classifier if it is not one of the pure disturbances from database The output of the classifier contains more than one disturbances if it is mixed in nature

The system uses two step tree structure In first step it decides whether there is pure disturbance or not If it is not a pure disturbance, it checks for all possibilities of mixed disturbances in second step In each node of a tree structure, it consists of fuzzy rules and yields fault measures at its leaves The architecture of Sequential classifier with input disturbance vector having four features is shown in Figure 4 17 Here, features THD (Total Harmonic Distortion) and DIH (Dominant Interharmonics) are considered as having mixed disturbance generating nature Figure 4 16 shows a simple example of such a tree structure In sequential classifier, user has to provide the information about the features which are having mixing nature He has to give information about the disturbances other than no disturbance based on that feature he has to provide information about the normal voltage with no disturbance Every feature has different ranges for different disturbances We have assumed that all feature must have one and

only one range which is for normal waveform It defines *no disturbance*
Here f is membership value after fuzzy inference rule operation on the unknown

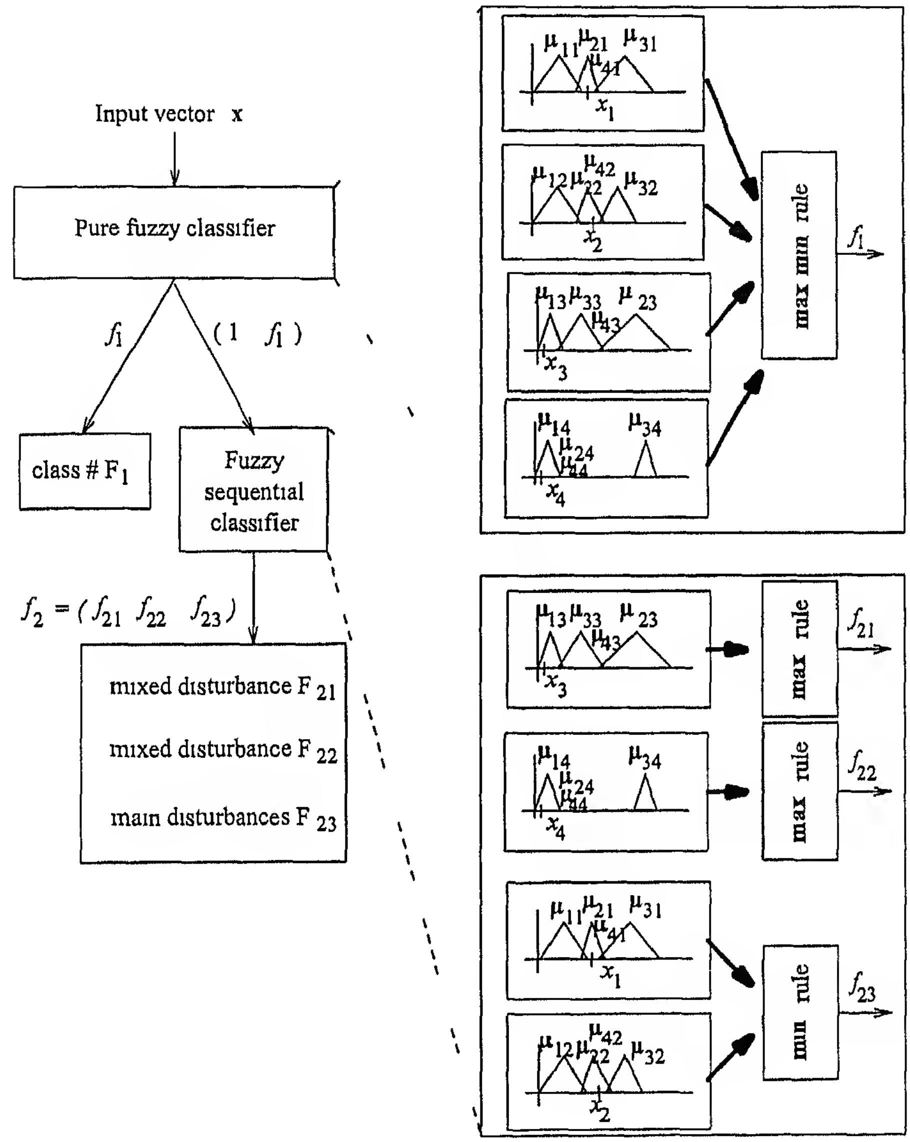


Figure 4 16 Classification tree structure for mixed disturbances in sequential classifier For reference, please refer in Appendix [5]

input vector x As shown in the tree structure f_1 is always a membership value resulted by fuzzy pure classifier which includes all disturbances with all features

of the vector The fuzzy pure classifier uses Mamdani $max - min$ inference rule for decision making class F_1 is a pure disturbance if $f_1 > 0$ Otherwise it will proceed for the sequential classification of using information about the features having mixed nature It finds the mixed disturbance based on each mixed feature individually using max inference rule on that particular feature The output of this step is a vector of membership values can be shown as f_2 f_{2j} is a membership value after max rule operation on j^{th} feature having mixing nature It also checks the possibilities of main disturbances if there is any feature which doesn t have mixing nature Here, it uses all features which doesn't have mixing property together like fuzzy pure classifier and min operation on every disturbance It results a vector of size equal to number of disturbances and consists of one membership values for every disturbance after min operation In the figure of tree structure the vector is shown by f_{23} The value of vector for i^{th} disturbance say f_{23} if positive then it may be the main disturbance which could be mixed with the other disturbance found previously Normally, only one disturbance out of all has positive membership value in the last case If there are more than one disturbance in this main disturbance detection case then it is not perfect detection In that case user may reselect the features having mixing nature by providing more features having mixing nature

The pure disturbance classification is same as $max - min$ principle used previously in fuzzy pure classifier Fuzzy mix classifier uses max rule for detecting mixed disturbance from features having mixing nature, for j^{th} feature it is stated as follows

$$
\begin{aligned}
S^{j}(\mathrm{x}) &= arg\, max_{=1}^{di_j(cd(j))} \mu_{\ j}(x_j) \\
\text{Where} \quad F_{2k} &= S^{j}(\mathrm{x}) \\
\text{and} \quad f_{2k} &= max_{i=1}^{c} \mu_{ij}(x_j) \\
\text{Where} \quad j &= index(k), \quad \forall k = 1, \quad\quad , c_{m\ x} \\
\text{Where} \quad c_{mix} &= \text{number of features having mixing nature} \\
\text{and} \quad cd(j) &= \text{number of disturbances accompanied with } j^{th} \\
&\quad\ \text{feature including no disturbance} \\
\text{where} \quad di_j(i) &= \text{disturbance index of } i^{th} \text{ disturbance accompanied} \\
&\quad\ \text{with } j^{th} \text{ feature of mixing nature} \qquad (4\ 31)
\end{aligned}
$$

For detecting the main disturbances out of features which are not having mixing nature, the formula is stated in Equation 4 32 Flow chart of fuzzy sequential

classifier during testing stage is shown in Figure A 6

$$\begin{aligned} S_i^{main}(\mathrm{x}) &= \omega_i \quad if\ \mu_i(\mathrm{x}) > 0 \\ \text{Where} \quad \mu_i(\mathrm{x}) &= min_{j=1}^{n} \mu_{ij}(x_j), \\ \mu_{ij}(x_j) &= 1 \quad \text{if } j^{th} \text{ feature has mixing nature}, \\ F^{i}_{2(m+1)} &= S^{main}(\mathrm{x}), \\ \text{and} \quad f_{2(m+1)} &= \mu(\mathrm{x}) \end{aligned} \tag{4 32}$$

As shown in Figure 4 17 fuzzy sequential classifier is designed to classify mixed

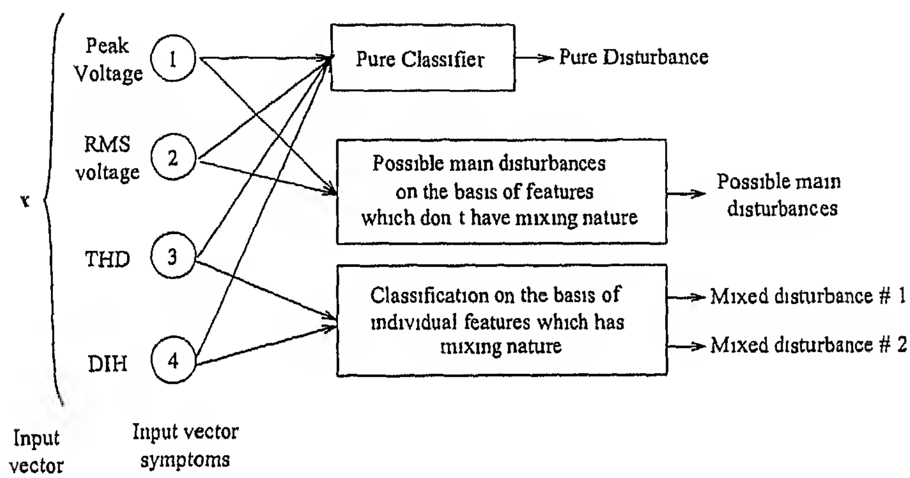


Figure 4 17 Fuzzy sequential classifier architecture

disturbances sequentially if it is not a pure disturbance Here, features THD (Total Harmonic Distortion) and DIH (Dominant Inter harmonics) are selected as having mixing nature So, they are classified individually using max function So, possible output of this step will be only one disturbance for every feature If the disturbance content of that feature is normal it classifies it as *no disturbance* For the feature THD, the disturbance resulted may be *Low Harmonics* or *High Harmonics* if it is not a normal value of that feature Similarly for DIH, the result may be *Inter harmonics* if it is not normal value If the value is unknown for that feature's fuzzy membership functions then the result will be *unknown disturbance* After that, the classification on the basis of features which don't have mixing nature is proceeded Here, the classification is on the basis of may be more than one feature but it doesn't include features having mixing nature So some disturbances which depends on some features which don't have mixing nature and features which have mixing nature both, i e Oscillatory Transients is having low harmonics content which is a dependent on feature THD and THD is selected

as a feature having mixing nature and impulsive transient which only depends on the features which don't have mixing nature here So, when classifying on the basis of features which don't have mixing nature there is a possibility that more than one disturbances resulted The final disturbance is considered as one main disturbance mixed with the disturbances resulted from individual features having mixing nature

5 Classifiers Testing and Results

5.1 Test Procedure

After having idea about the generation of PQ disturbance data in Chapter 3, we have to test the classifier to evaluate their performances for PQ classification problem. For that we should test the classifiers with different data distributions over feature ranges as the real world data distributions are not uniquely defined. In the following points, we mention the tested data distributions and generalised test procedure.

5.1.1 Testing with different data distributions

We have tested the classifier with following three types of possible data distributions.

a) **Uniform distribution** Data is distributed over all feature ranges uniformly. Mathematically generation of i^{th} feature of a vector x^j for j^{th} class is stated as follows

$$\begin{aligned} \text{dataset } x_i^j &= min(x_i^j) \\ &+ (max(x_i^j) - min(x_i^j)) * rand([0,1], N) \end{aligned} \quad (5.1)$$

Where N is number of data points to generate and $rand[0,1]$ represents the uniformly distributed randomly data generator function from 0 to 1.

b) **Normal distribution** It is also known as Gaussian distribution. Generation of the data of this distribution is same as of uniform distribution except that the data generator is normal distribution function.

c) **Abnormal distribution** The data is generated by randomly dividing the feature range with random subpart of the total data to generate in each sub part of the feature range. Each subpart of the feature range having normal distribution. Here the data distribution shapes of different features for the same and different type of classes are not same but random.

5 1 2 Generalisation of testing procedure

The common procedure adapted for testing of all classifiers is described as follows

Training set and test set

The data set is splited into two parts before used for classifier evaluation The first part is used to design or construct the classifier called the *Training Set Learning Set* or *Knowledge Data set* and the second part is used to evaluate the performance of the classifier called the *Test Set* The dataset is divided into two parts sequentially after generating it randomly As a general way the first 60% data is used as training set and rest is used as a test set It s better to have a large dataset for classifier performance evaluation Generally we used to have a dataset minimum of 1000 data vectors for every class If we use the training set for classification then it s called *reclassification* The reclassification would be only for the proof of mathematical functionality of the classifier

Training process of the classifier is shown in Figure 5 1 Training set has a set of pairs of data vector and it's class of belongingness As shown in a figure mentioned above, classifier is trained to classify the unknown test vector from using the knowledge of the class type of data vectors in training set In Figure 5 2, testing procedure of the classifier is shown In this stage classifier is assumed to have a capacity to classify the data samples from training set Here, number of test vectors of every class which are correctly classified by it's own class misclassified by some other classes or becomes unknown are calculated from using the knowledge about the class belongingness of every data vector of test set Generalisation procedure of different classifiers are shown in figures attached in the appendix in the form of flow charts The results of the classifier is presented in form of error matrix which is explained in the next section

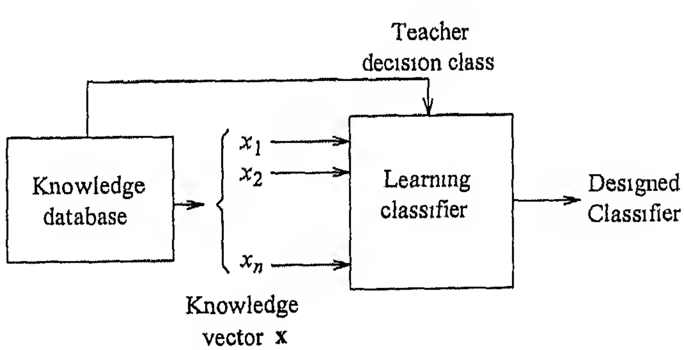


Figure 5 1 Classifier Training Process

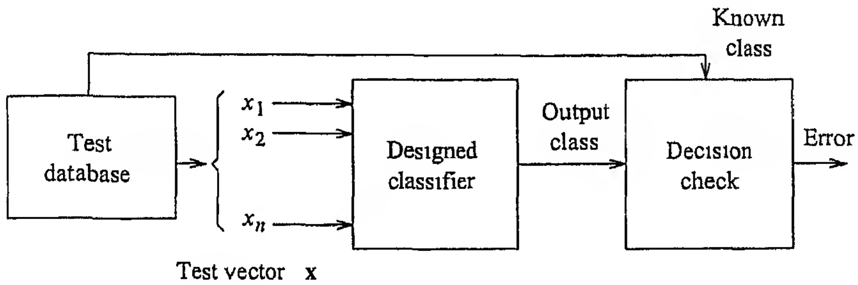


Figure 5.2 Classifier Testing Process

Error matrix calculation

Error matrix H is a result matrix generated on the basis of classifier performance on test data. It is a square matrix and the size equal to number of classes $(k \times k)$. $H_{ij} \in [0, 1]$ indicates the value in per unit of number of classification of i^{th} class test vectors resulted in j^{th} class. It can be represented in % also. Generally, diagonal values are much higher than off diagonal values of the matrix H as it represents the correct classifications. If, suppose for i^{th} row, any off diagonal value is higher than diagonal value of the matrix in that row or the diagonal value is comparatively lower than other diagonal values then classifier is said to be poor to classify the vectors of i^{th} class and misclassifying the vectors of that class in much more amount. The graphical representation of error matrix is called *Histogram chart*.

The accuracy of the classifier can be calculated on the basis of number of misclassified vectors out of tested vectors in every class. If the vector is classified as unknown or classified in wrong class then it is said as misclassified vector. The accuracy of the classifier is defined as follows

$$\text{accuracy} = \left(\frac{\text{number of correctly classified vectors}}{\text{number of tested vectors}} \right) 100\,\% \tag{5.2}$$

Development of test vectors

We have considered here six disturbances having six features for general testing of all classifiers first. The *lower limit* and *upper limit* for every feature of every disturbance is shown in Table 5.1. The total ranges of all features are normalised and individual features' ranges of all disturbances are calculated linearly on that basis. Duration range is original in milliseconds and rise time range is normalised to 20. The data generation for this disturbance-feature set can be done in 1 per unit or 100 % normalised scale. As they are related linearly, just linear multiplication factor converts scaling from per unit to linear and vise versa. Normally,

	V_m	V_{RMS}	T_e ($\log_{10}$)	t_r ($\log_{10}$)	THD	DIH
Impulsive Transient	0 3 1	0 2 0 3	0 5 5	0 1 2	0 0 2	0 0 1
Oscillatory Transient	0 3 1	0 2 0 8	0 5 10	0 1 5	0 0 5	0 9 1
V Sag	0 0 18	0 0 18	5 100	0 1 20	0 0 2	0 0 1
V Swell	0 22 0 28	0 22 0 28	5 100	0 1 20	0 0 2	0 0 1
Harmonics	0 18 0 22	0 18 0 22	5 100	0 1 20	0 2 1	0 0 1
Inter harmonics	0 18 0 22	0 18 0 22	5 100	0 1 20	0 0 2	0 9 1

Note

V_m	Peak voltage	V_{RMS}	Voltage magnitude
T_e	Duration of Event	t_r	Rise time
THD	Total Harmonic Distortion	DIH	Dominant Inter harmonics

Table 5 1 Normalised Feature ranges of six disturbances with six features by [0 1 pu] This ranges are used for different experiments on classifiers to decide the the optimal feature ranges and distributions of the data points Here, duration feature is shown in ms and rise time is shown in a range of [0 20] This data range structure is used for testing different classifiers

duration and and rise time are redistributed with logarithmic function after generating so their ranges would become comparatively symmetrical to other feature ranges The reason for this will be mentioned in the next section The distribution histogram of all six individual features of all disturbances for uniform normal and abnormal distributions are shown in three different figures attached in the appendix

5 2 Test of Individual classifiers

5 2 1 Euclidean Minimum Distance Classifier

This is the first classifier we implemented for PQ classification problem Because of it s simple structure for analysis, we did different tests for the data generated from Table 5 1 We found some basic structure of data generation on the basis of these tests which described as follows

Visualisation and analysis of error matrix

The data set generated from feature ranges as shown in Table 5 1 is used for testing the Euclidean classifier Here, duration and rise time values are redistributed using $\log_{10}$ function to provide the uniformity of feature ranges for different classes The results of generalisation is shown in Figure 5 3 It can be observed that the error matrix for first class belongingness as shown first from left in the figure above has good accuracy
Here three error matrices are shown in the figure Second and third from left show second and third class of belongingness Every row shows results of that row class samples The height of every bar indicates the fraction of total test samples from the row class classified by the class indicated by the class index below it Second and third belongingness results provides the additional information about the classes surrounded by the main class
The results without using logarithmic functions are shown by a figure attached in the appendix It is not having good accuracy due to uneven width of duration and rise time for different disturbances It is also found that linear transformation of the ranges will not affect the results of Euclidean distance classifier The trials were done to do nonlinear transformation of feature ranges to improve the results The results for some feature ranges changed for uniformity of feature ranges for different disturbances are attached in the appendix It is found that results of Euclidean distance classifier is affected by the nonlinear transform of feature ranges of different classes The shifting of boundaries will result into change in second and third class belongingness can be observed in the same figures with nonlinear transformation or mapping of some features
The test results with abnormally distributed data is shown in a Figure in the appendix It is found that the performance of the classifier becomes less accurate if the data is abnormally distributed over the feature ranges

Pros and cons of the classifier

From the testing results above, we can describe the pros of Euclidean classifier as follows

- It has simple design structure to understand
- Easy to develop the classifier
- It has less memory requirement
- It has less testing time

The cons of the classifier are listed as follows

- It has poor accuracy

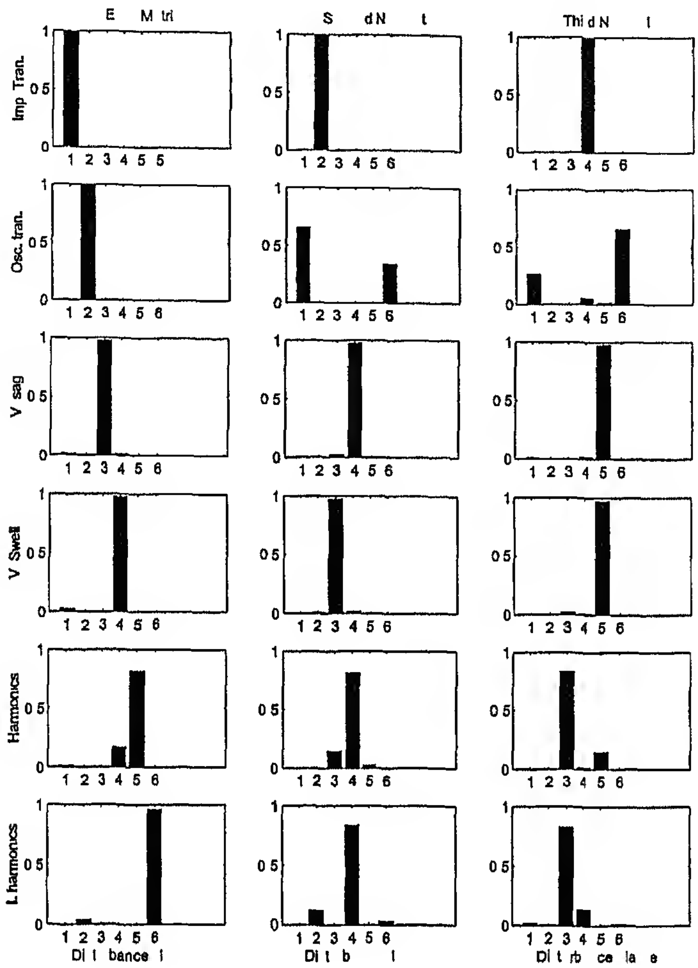


Figure 5 3 Euclidean distance classifier error matrix with total range is [0 1] and *rise time* and *duration* are redistributed with $\log_{10}$ Data samples for every class is 1000 out of which 60% is used for training and 40% is for testing The originally data of all features are distributed uniformly The results are improved very well compared to that in Figure A 7 So, it is must here to redistribute the data of duration and rise time features by logarithmically to improve the results The second and third columns provides the results in form of error matrices for second nearest and third nearest classes as mentioned in Figure A 7

- It depends on only one vector called the mean vector for every class, so if the feature ranges of particular disturbance is uneven then it is not desirable

for good results

- Hyperplane decision surface location depends on the distance between neighbouring disturbance classes mean vectors Due to nonuniform distribution of mean vectors of different classes in the input space decision surface of a particular class is not equally distant in all directions from it s mean vector This may lead to misclassification of some of pattern vectors of that class

- Due to uneven volumes of training vector clouds of different disturbances it may possible that some test vectors of the class having large volume may fall onto the other side of the decision boundary This will result into the misclassification of test vectors

- As it works on the minimum distance and it just comparative measure So If the input test vector is unknown and doesn t belongs to any of knowledge classes then also it will misclassify by resulting it to any one of the knowledge classes

- The class representative is only one vector So, it gives unexpected results during testing if the data distribution for every class is not symmetrical around the mean vector

5 2 2 Bayes Classifier

As we have seen some tests on Euclidean distance classifier to improve the data distributions, now onwards we will consider the data of 6 disturbances with 6 features having normalisation by [0 1 pu] range on linear scale only as shown in Table 5 1 and duration and rise time are redistributed with $\log_{10}$ function The data range structure is shown in Table 5 1 We will test the classifiers with all three types of data distributions The results after testing the Bayes classifier are discussed as follows

Visualisation and analysis of error matrix

Figure 5 4 shows the error matrix for the data mentioned above with uniform distribution of 1000 data samples of every class The representation of error matrix is same as in Euclidean classifier for first class of belongingness We can observe that the error matrix histogram is almost diagonal means that all the class test samples are almost classified in their own class The number of vectors misclassified are very less here The accuracy of classification is above 99 % for all classes So the results are better than any results with Euclidean distance classifier as seen in Section 5 2 1 Here, the results are based on the Bayes probability value for different classes for the particular input sample So there is probability value which follows Gaussian distribution principle for the

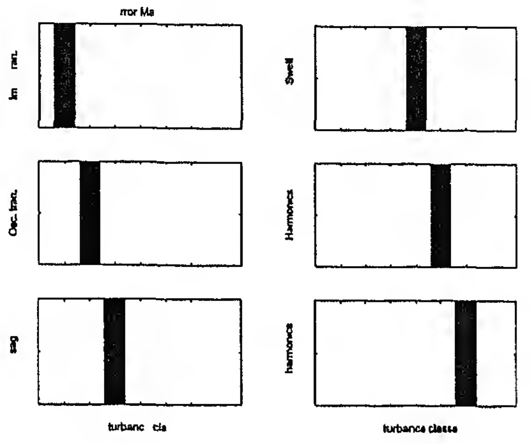


Figure 5 4 Bayes classifier error matrix with data number data distribution feature range and individual class feature ranges are same as in Figure 5 3

cluster of a particular class So if the sample far from the cluster then it s Bayes probability of belongingness in that class would be very less If the sample belongs to that particular class cluster then the Bayes probability of belongingness in that class is comparatively much higher than other classes So, there is no possibility for finding the second class of belongingness in case of Bayes classifier unlike Euclidean classifier as we have seen

We have tested this classifier with all classes having *a priori* probabilities same (maximum likelyhood) If it is considered then the only the multiplication factor to the Bayes probability function will change for different classes and the variance and the mean values will not be affected So, if two classes are overlapping then this principle of *a priori* probability will work nicely The sum of *a priori* probabilities of all classes is always unity

Pros and cons of the classifier

The pros of Bayes classifier can be stated as follows

- The result are having very good accuracy It can be used as a reference measure of accuracy for any other type of classifier since it provides almost ideal results

- It works on the principle of *a priori* probability It is useful for our case because, in real power system different events are not having the same frequency of occurrence e g some survey in different parts of the worlds

says that Voltage Sag is having the highest frequency of occurrence So, the *a priori* probability can be calculated for every class on the basis of their frequency of occurrence and can improve the results

- It is fast enough for testing large number of data
- It calculates the Bayes probability using covariance matrix, so if the variance of different feature data for the same class is not same then also it doesn t affect the results unlike in Euclidean distance classifier
- The Bayes probability value for every class depends only upon the distribu tion of it s own class data samples So, the orientation of other neighbouring class data samples doesn t affect the results unlike in Euclidean distance classifier
- Sizes of the different class clusters don t affect the results here
- Bayes probability value is very less if the sample is outside the class cluster So it is possible to limit the misclassification of unknown class sample by any known classes if the sample doesn't belong to any one of the known classes
- The results improve if the data samples are large enough

The cons of Bayes classifier can be stated as follows

- The Bayes probability works on the basic principle of Gaussian distribu tion If the data samples of any class are not distributed according to Gaussian distribution, the Bayes probability function may become having wrong mean value and covariance matrix which may lead to misclassifica tion This case is dangerous when the data samples of different classes are just touching and the distribution over the feature ranges are uneven or abnormal The real power system events data is not Gaussian distribution for all features and even it varies from class to class for the same feature The real system data results' accuracy may decrease from this results For example, one abnormal case is shown in Figure A 1 So, results becomes less accurate if the data is distributed towards the corners of some features
- It doesn't work well when the classes are overlapping
- It has smooth boundaries around the mean vector, so for data at the end limits of features may lie on the corners may misclassify if the another class data samples adjoining the border are present

5 2 3 k-NN Classifier

We have tested k NN classifier with the data generated from the data ranges shown in Table 5 1 We have tested with k=1 and k=10 nearest neighbours for 1000 data samples in dataset for every class and compared the results The results are discussed in the next section This classifier has some additional advantages compared to other classifiers in PQ classification system are also discussed It is tested with different type of data distributions discussed in Section 5 1 1 and results are compared

Visualisation and analysis of error matrix

The results for uniformly distributed data and normalisation of feature data ranges by [0 1pu] as shown in Table 5 1 and taking $\log_{10}$ to duration and rise time features are shown in Figure 5 5 The first and second columns from left shows Error matrices for results of 1 NN and 10 NN classifiers respectively It can be observed that results of 1 NN classifier is slightly better than that of 10 NN classifier Both results are better than Euclidean distance classifier results for the same data as shown Figure 5 3 As mentioned in Section 4 3 k NN classifier provides piecewise linear classification, it improves the nonlinearity in the decision boundaries So, it is having more capabilities to classify if the data samples of neighboring classes are very near to each other If the number of nearest neighbours is decreased (value of k), the nonlinearity of decision surface will increase

We can also see in Figure 5 5 that third and fourth from left represents the error matrices for second nearest class and third nearest class for 10 NN classifier It can be calculated from the selected k neighbours for every test sample So these information is not possible in case of 1 NN classifier because it has only one neighbour for class information If the test sample is inside the cluster of the class data samples then class data samples then almost all k neighbours will have a same class belongingness but if the test sample is on the boundaries of the class cluster then it may possible that it will have k neighbours with different class belongingness In that case some neighbours out of k neighbours from neighbouring class clusters also present In that case one can have an information about the second belongingness of the test data sample So, it provides the probabilities of belongingness of the test data sample in different classes It is called the information about classification quality No other classifier provides this type of information So, this is the unique feature of k NN classifier From figure, one can observe the histogram chart of second class and third class belongingness for test samples of every class in per unit values Number of test samples for every class is the base values for all graphs For first row results of Impulsive transients test samples, there is no information in third and fourth columns, it is because there is no neighbour from any other class except the one class in which the test

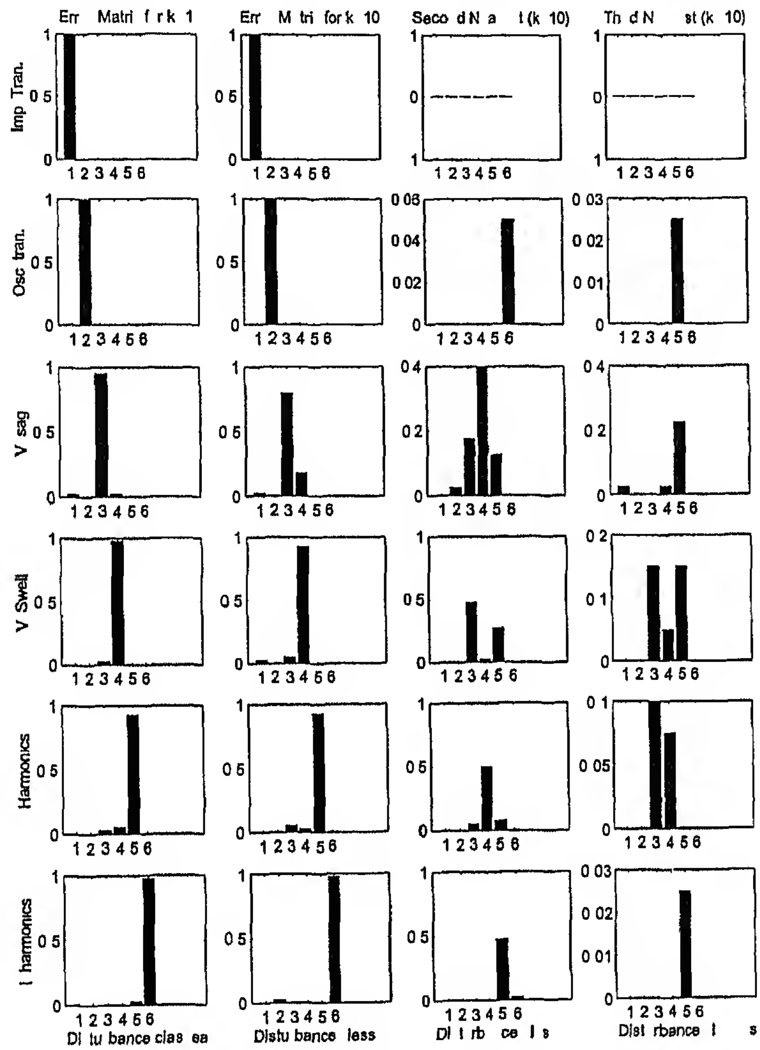


Figure 5 5 k NN classifier error matrices with data distribution, feature range and individual class feature ranges are same as in Figure 5 3 100 data samples for every class out of which 60% is used for training and 40% is used for testing The structure of four error matrices is same as in Figure A 11 The results for k=1 and k=10 NN are almost same It is slightly better than Euclidean distance classifier as seen in Figure 5 3 It gives information about class clusters orientation as discussed in Figure A 11

sample belongs out of k neighbours For oscillatory transients, the number of test samples is very less which has second or third class of belongingness
The results of k NN classifier tested with abnormal data is shown in Figure A 11 Here it is observed that when the data is abnormal then the results with 1 NN and 10 NN becomes less accurate the possibilities of misclassification in neighbouring class will increase when the class cluster is having variation of density of trained data points throughout the cluster So it would be better to have uniform knowledge data points distribution for the every class cluster to get good results of k NN classifier As shown in figure, particularly voltage sag and harmonics are misclassified in neighbouring class of voltage sag and each other The addition information of classification quality as mentioned in last paragraph is useful here

Pros and cons of the classifier

The pros of k NN classifier can be stated as follows

- It has better accuracy than Euclidean distance classifier
- The classifier structure is very simple to understand
- It provides the additional information about the classification quality
- The effectiveness of the classifier can be tuned by changing number of nearest neighbours (value of k)
- It is capable to classify the class clusters which are overlapping or one inside the another class clusters or clusters in the form of sandwich
- The results are not affected by the shape or size of the cluster
- Training of the classifier is very simple and improvement of the knowledge database to modify the performance of the classifier is simple

The cons of k NN classifier can be stated as follows

- Accuracy is not good compared to Bayes classifier
- It requires huge storage for the knowledge data points
- It is very slow due to calculation of the Euclidean distance with each and every knowledge data point from test point
- The results are affected by the uneven distribution density of different neighbouring class clusters' knowledge data points
- If the distribution of the data points in a particular class is abnormal then misclassification may increase

5 2 4 Fuzzy Classifier

As shown in Figure 5 6 the error matrix of fuzzy classifier for uniformly distributed data is almost diagonally predominant There is rarely any misclassification found by fuzzy classifier The results are very good out of all classifiers tested here The results may become bad if there is abnormal data distribution The membership function is used here is Equilateral triangle for every feature of all classes The mean value is calculated from the mean of the data points and the base of the triangle is calculated on the basis of maximum width on any side of the mean value of the feature range The result doesn t change if the membership function changed by trapezoidal hyperbolic parabolic circular or some other shapes with the limits remains the same It is also tested with Gaussian membership functions the results are almost same with the other results of fuzzy classifier It can be recognized that it has almost same functioning as Bayes classifier Here we have some flexibility for membership function generation unlike in Bayes classifier

As we have discussed during feature selection stage in Section 3 4 1 that the

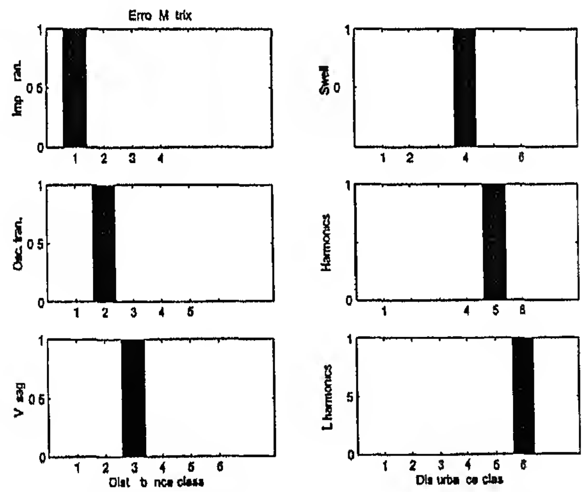


Figure 5 6 Fuzzy classifier error matrix with data number, data distribution feature range and individual class feature ranges are same as in Figure 5 3

features are selected such that their values are identity of the particular class So, it is good to apply Fuzzy classification strategy developed here based on the membership function for every feature of all disturbance classes So this strategy can be used to detect mix disturbances by step by step classification based on individual feature values of the unknown or test sample This will be evaluated

in section 5 2 6

Pros and cons of the classifier

The pros of Fuzzy classifier can be stated as follows

- Accuracy is very good compared to all other classifiers
- Simple to understand and modification for knowledge improvement is easy
- Flexibility to select different type of membership functions
- Testing is fast enough
- Feature by feature classification in steps can be developed easily So it can be used for sequential classification
- Less storage required
- According to power system disturbance data generation and ranges creation, fuzzy classifier strategy is good enough for classification here
- It classifies the test sample which doesn t belongs to any class as *unknown class* instead of misclassifying by some other class
- Only the minimum and maximum limits of every feature of all disturbances are enough to construct the classifier

The cons of Fuzzy classifier can be stated as follows

- Results becomes not good if the classes are overlapping or one surrounds the other
- More than one membership function for any feature of the class is not allowed according to classifier strategy we used here
- If the min and max limits are not true enough then the possibilities of misclassification increases This may be one case of abnormal data distribution It works on mean value and limits of the feature data points, so, abnormality of the data points is not desired Data symmetry on both sides of the mean value is desired

5 2 5 RCE Classifier

The results of testing RCE classifier is shown in Figure 5 7 The results are almost good accurate compared to Euclidean distance classifier and k NN classifier

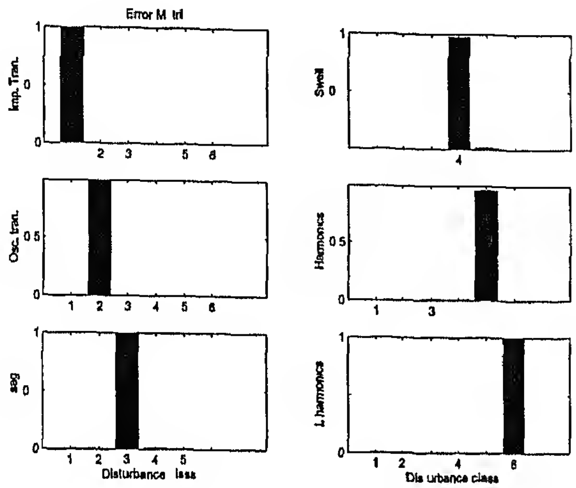


Figure 5 7 RCE classifier error matrix with data number, data distribution, feature range and individual class feature ranges are same as in Figure 5 3

Pros and cons of the classifier

The pros of RCE classifier can be stated as follows

- It has good accuracy
- It is easy to train the RCE classifier at any stage
- Less storage required
- Fast testing
- It can classify the test sample which doesn't belongs to any class as *unknown class* instead of misclassifying by some other class
- It doesn't affected by the uneven size of different class clusters
- It can classify the classes separated by nonlinear decision boundaries
- More training improves results
- Data distribution doesn't affect the results

The cons of RCE classifier can be stated as follows

- Training is slow

- It fails when the clusters overlap
- It is difficult to get high accuracy like Bayes classifier

5 2 6 Fuzzy Sequential Classifier

Fuzzy sequential classifier has been tested for the mixed disturbances data The data generation of mixed disturbances is same as for the data of pure disturbances The change here is only the ranges of the different class test vectors according to the type of mixed disturbances The knowledge dataset will have a samples of pure disturbances and the test dataset will have the samples with mixed disturbances The number of output classes are predefined and includes all pure disturbances plus possible mixed disturbances

Visualisation and analysis of error matrix

As the results shown in Figure 5 8, it has good accuracy for pure as well as mixed disturbances classification It can classify 8 pure disturbances and 12 mixed disturbances Here only 3 disturbances results are shown for example It is accurate for any of the disturbance testing out of 24 different classes created from 8 basic classes The features used here are only four Two features duration and rise time are not selected here Here only fuzzy classification technique is used for sequential classification Some other classifiers could also be implemented for sequential classification

Pros and cons of the classifier

The pros of Sequential Fuzzy classifier can be stated as follows

- It can provide the information about the mixed classes and subclasses
- More detailed classification is possible
- fast enough compared to other classifiers

The cons of Sequential Fuzzy classifier can be stated as follows

- Structure is complex
- The disturbances could be mixed is decided by a particular feature So, there must be some feature which can detect the disturbance alone without using the information from other features
- Limited number of mixed disturbances can be classified

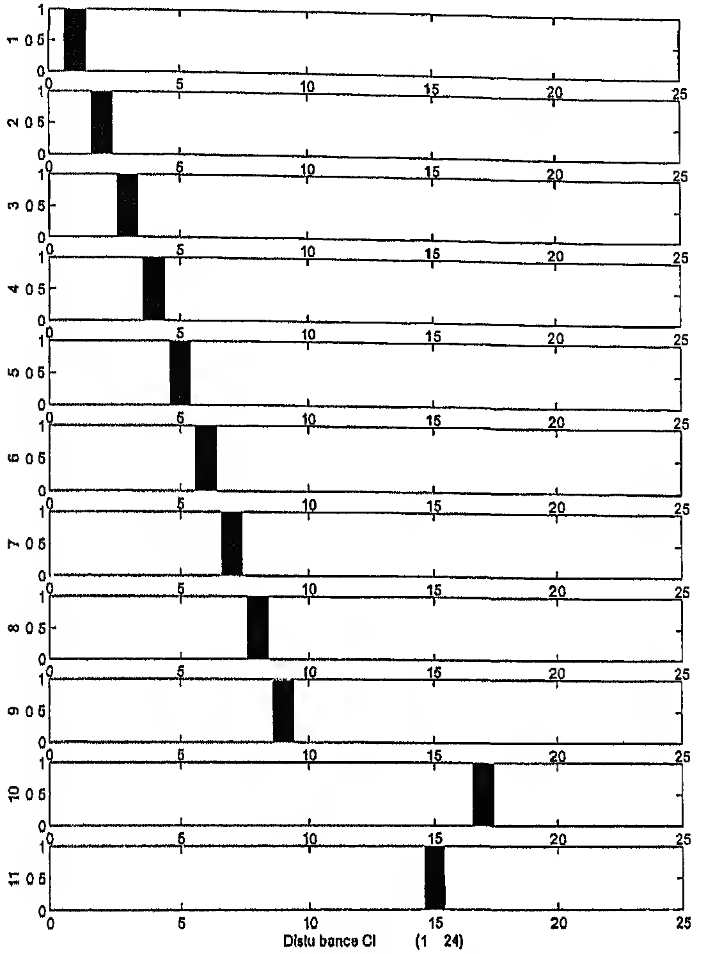


Figure 5 8 Sequential fuzzy classifier error matrix with eight knowledge classes and 11 tested classes using four features It shows that fuzzy se quential classifier classifies mixed as well as pure disturbances with good accuracy The pure disturbances in row sequence are stated as follows 1 Imp Tran , 2 Osc tran , 3 V Sag, 4 V swell, 5 Low Harmonic, 6 High Harmonic, 7 Interharmonic 8 No Disturbance, 9 Imp Tran + Low harmonic 10 V Sag + Interharmonic 11 V Swell + High Harmonic 24 columns of error matrix defines 8 pure disturbances and 12 mixed disturbances

5 3 Total Results

So far the classifiers have been tested individually with different data due to the state of the development progress Now, all classifiers are trained and tested

under equal conditions The same data set of 7 disturbance classes and 4 optimum features is used for evaluation of generalisation capacities of all different classifiers The results are shown in Table 5 2 Here, we have not shown the results of Sequential classifier it is because it functions as same as fuzzy classifier for pure disturbances It differs only when the test data has mixed disturbance samples We can yield to the decision on the basis of these results and pros and cons of every classifier discussed in the last section that the Bayes the k Nearest Neighbour and the Fuzzy classifiers together are best for our implementation work In the next chapter we will try to show User Interface Program for implementation of PQ disturbances classification with some examples

Disturbance type	Euclidean classifier	Bayes classifier	Fuzzy classifier	Euclidean k Nearest Neighbour c	RCE neural network c classifier
no disturbance	100%	100%	100%	100%	100%
impulsive tran	78 800%	100%	100%	100%	99 975%
oscillatory tra	78 800%	100%	100%	100%	99 875%
voltage sag	99 575%	100%	100%	100%	98 5%
voltage swell	100%	100%	100%	99 875%	96 875%
harmonics	81 9%	100%	100%	100%	87 775%
inter harmonics	100%	100%	100%	99 850%	99 975%

Table 5 2 Results of Generalisation test of 4000 uniformly distributed test patterns for each disturbance, after training with 6000 sets for each type of disturbance Every class has four optimum selected features

6 Realised Classifier

In this chapter, the realised classifier system is presented The system is implemented in Matlab[11] There are various GUI (Graphical User Interface) for modification and result assessment of the classifier

6 1 Basic Structure of parallel classifier

The simple structure of parallel classifier design using GUI system is shown in Figure 6 1 It works on the principle of parallel classification of unknown input vector using the Bayes, the fuzzy and the k NN classifiers The user can generate a new database or use a previously generated database He can select the test vector or create a new one The results assessment from classification system to the user is done The three results of three different classifiers are assessed to the user The voting of all classifiers is disclosed for the unknown input vector class identification There are two possibilities of results First, the results of all three classifiers are agree then nothing to do If the result of any classifier is disagree with other classifiers then the flexibility to train the classifiers is provided for that unknown vector The user can add a vector to the database when he doesn t satisfy with the results and wants to improve the database We developed GUI system for implementation of fuzzy sequential classification system for mixed disturbances as well as pure disturbances classification The main steps of GUI system developed in *Matlab* are explained as following

6 2 Database Creation

The control window for Database creation window in matlab platform is shown in figure 6 2 Here, first user has to input number of classes and number of features, applying *OK* there will provide the new window as shown in above figure to fill up the different parameters required to generate the new database The figure shows the filled up window for 8 disturbances and 4 features User has to provide number of knowledge data for every class under the filled name for every disturbance as shown in figure He has flexibility to choose the type of distribution for every feature e g normal, uniform etc and type of redistribution whether to use logarithmic scale or linear scale for every feature under the filled

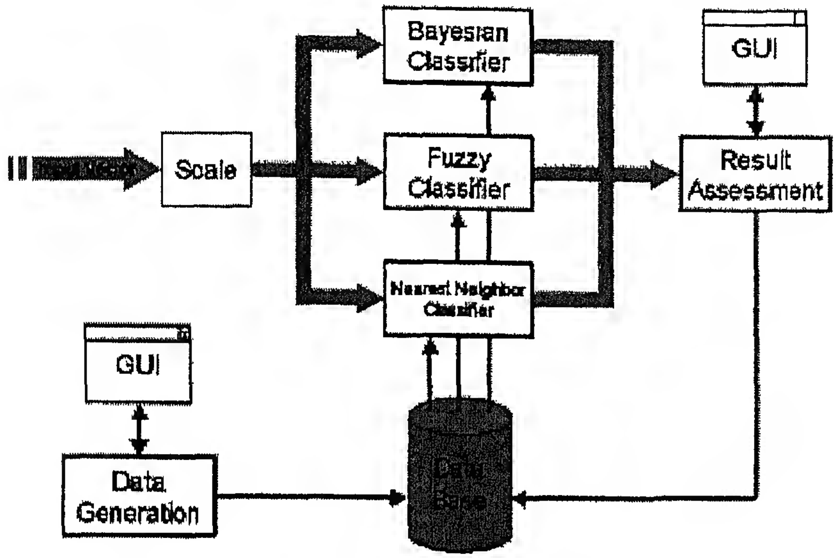


Figure 6 1 Basic structure of realised parallel classifier

up feature names The he has to fill up minimum and maximum limits for all features of every disturbance as shown in the figure The scale represents a linear scaling factor from real data to the classifier input data After completing this process, user has to click on the *create data* button will ask for the file name for this new created database It will generate the data vectors according to the information filled up and store the whole structure to the new data file in Matlab

6 3 Testing and Result Assessment

The Matlab window for testing the parallel classifier is shown in Figure 6 3 First of all user has to select the data base before testing He has to click on *From database* button to select one of the created data bases Now he can generate the unknown vector by clicking on *Generate* button and filling the vector values Or he can select one vector from the data base for testing purpose He can choose the disturbance class of the test vector and the type of data base i e *test data base* or *knowledge data base* After that, he has to click on *choose* button to select one new test vector of specified class and data base Here, later the test vector of recorded signal will be inserted automatically online and classified Once the input vector is ready in the vector box, he has to just click on the *Classify* button to assess the results of the parallel classifier system

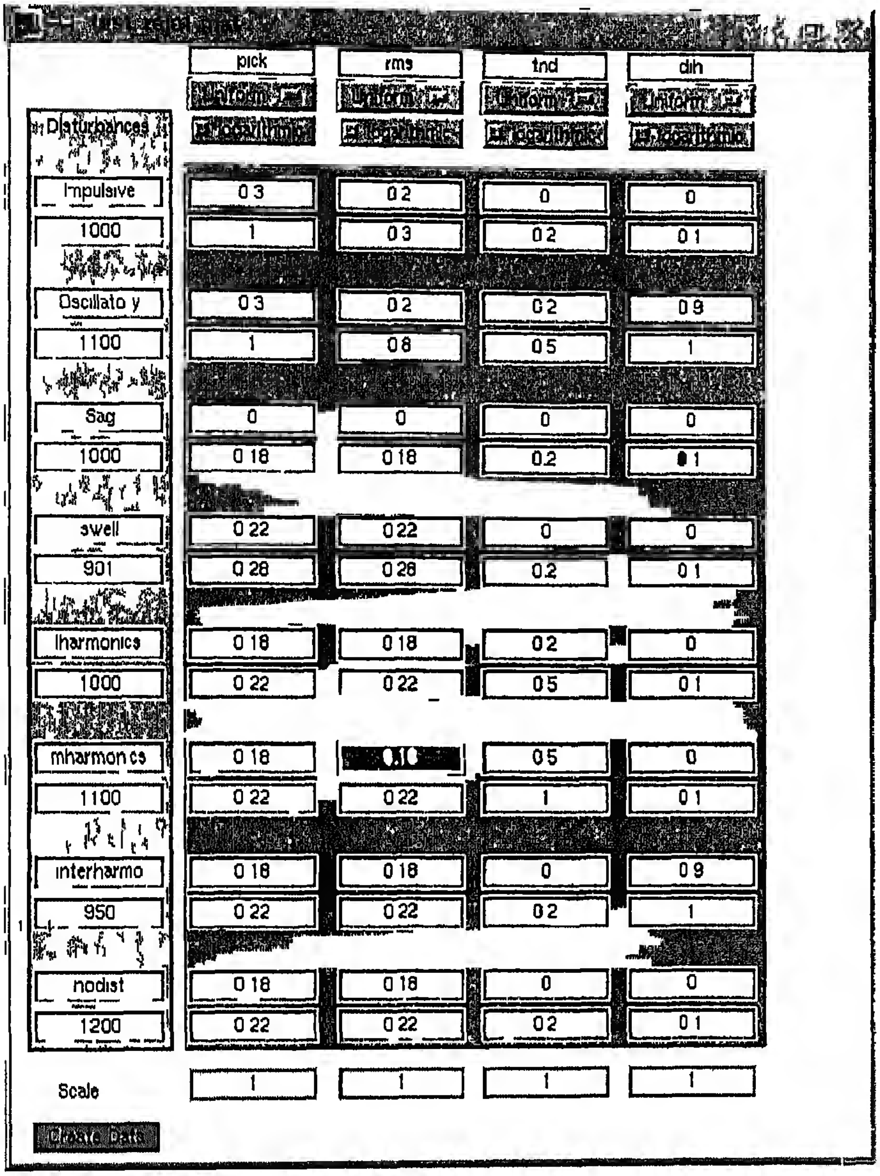


Figure 6 2 User Interface Program artificial database generation window

As shown in the figure, one input vector of *swell* disturbance is chosen from test data base The results shows that fuzzy, Bayes 1 NN and k NN classifier have the same result and all classifies the unknown input vector correctly The k NN classifier provides an extra information about the neighbouring class or the

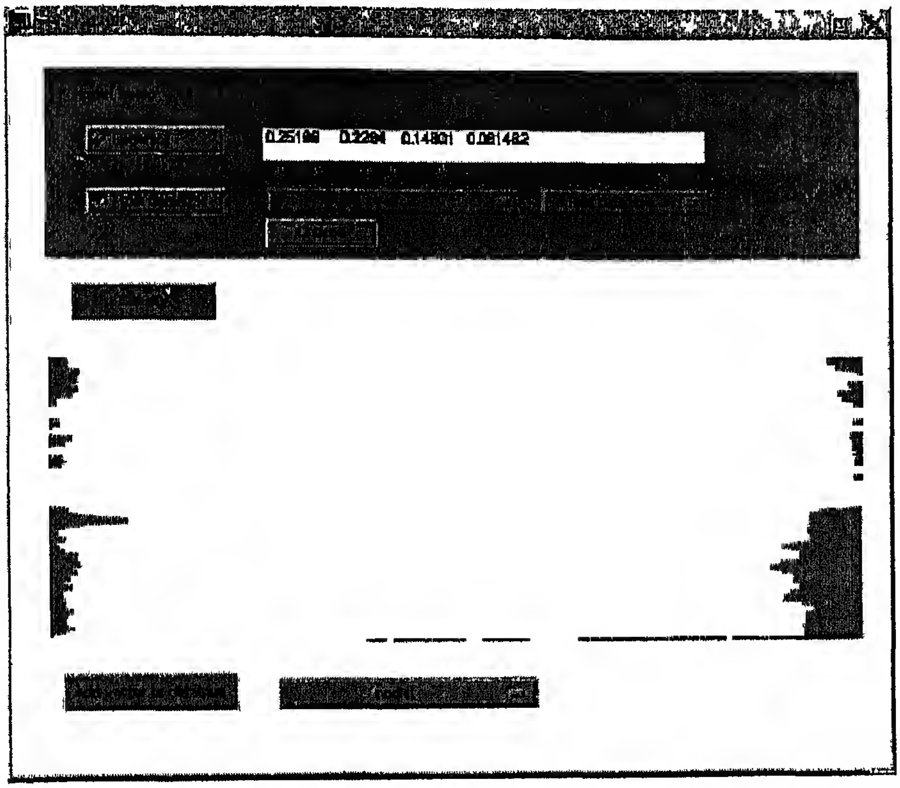


Figure 6 3 User Interface Program parallel testing of the unknown input vector

quality of the input vector Here it denotes that 99 % of the neighbours are from the same class of the input vector but only one vector is from *no disturbance* class It says that the input vector is having small swell which can be observed from the feature values of the vector displayed The second feature is RMS value in the selected database Here, results of all classifiers agree as swell So the disturbance is decided as swell If the results of all classifiers are disagree then the results are said to be wrong In that case, he has to see the waveform for the disturbance and decide the type of disturbance originally it was and modify the database to train all the classifiers for that vector So, if the same type of disturbance comes in future, hopefully it will be classified correctly by all the classifiers

If user wishes to add the tested vector to the database then he has to click on button *Add vector to database*, it will be asked whether he wants to add in particular class or not User can select the class in which he wants to add the new vector by selecting from pull down menu shown in figure Here it is no

disturbance class is displayed in pull down menu

6 4 Sequential classifier implementation

The Matlab window for *Mixed Classifier* is shown in Figure 6 4
As in parallel classifier here user has to first select a data base by clicking a

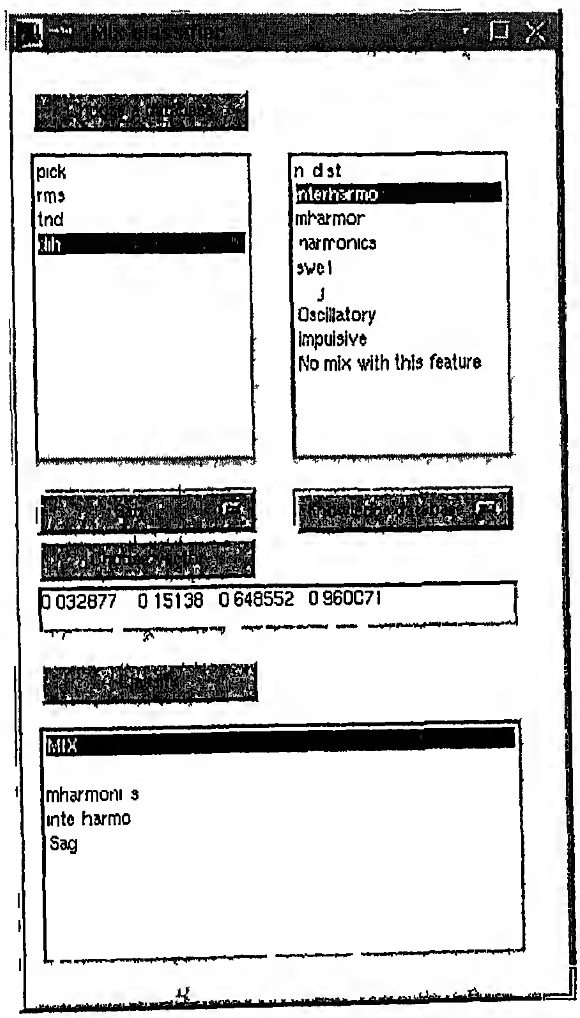


Figure 6 4 User Interface Program sequential classifier implementation

button *Choose a database* Now he has to establish a relation between features and disturbances He has to select the feature which has mixing nature in the

left column and the disturbances other than no disturbance which can occur due only that feature e g *dih* (Dominant Interharmonic) can create disturbance *interharmonic* mixed disturbance The selection is shown in figure And like wise *thd* (Total Harmonic Distortion) feature has been selected with *mharmonic* (more harmonics) and *lharmonic* (less harmonics) disturbances Here selection of the test vector is same as in parallel classifier discussed Or he can create the input vector by own by filling up in the vector box shown After that he has to click on the *Classify* to assess the results of fuzzy sequential classifier He will be asked to select the no disturbance out of all classes After clicking *OK* he will get the results of the classifier by more than one disturbance There will be one resulted class from every feature which has mixing nature and one or more from features which don t have mixing nature The information from every feature which has mixing nature identifies which type of mixed disturbance due to that particular feature if it is Otherwise it says there is no disturbance mixed due to that feature by identifying it as a no disturbance The result from rest of the features which don't have mixing nature, will identify the main disturbance

7 Conclusion

The focus of this project was to design a basic classifier system for power quality (PQ) events. We introduced the general pattern classification system and the principles of classification in Chapter 2. Here, the strategy for the design of pattern classification systems was presented. In Chapter 3 this procedure was adopted to PQ. Possible classes and features of PQ events were analysed. Out of them 6 signal parameters and 6 basic PQ disturbances were choosen for classification. We found out that 4 of the 6 features nearly consist of the same information by applying principal component analysis. The basic classification strategy of 5 different classifiers plus one with sequential structure are discussed in Chapter 4. In Chapter 5, we did the generalisation performance evaluation of 5 classifiers by changing feature ranges and data distributions to improve the classification performance and found that 3 classifiers out of 5 are suitable for PQ classification application. The realised classifier using a parallel structure of the classifiers with the best performance, includes the Bayes, the fuzzy and the k-NN classifier using various graphical user interfaces (GUI) is developed in Matlab. We also developed a GUI for Fuzzy Sequential classifier for classification of superimposed disturbances. The realised system is scalable that means additional disturbance types and different features can be added or changed. Also the system provides information about the quality of the classifier's decision. When unknown PQ events occure the system has the ability to expand its database. That is helpful to let the system identify a similar event correctly when it appears next time. Future development aspects include the design of an interface for a present data acquisition personal computer (DAQ PC) card to measure real data in power systems. Here the amount of disturbances to classify was set to 6 different types so this set can be extended to more disturbance types. Also the assessment should be completed with statistical evaluation of the classified events. In addition there are possible improvements for the classifier itself, e.g. the fuzzy sequential classifier for adapting more flexibility to classify various kinds of mixed disturbances.

A Additional Figures

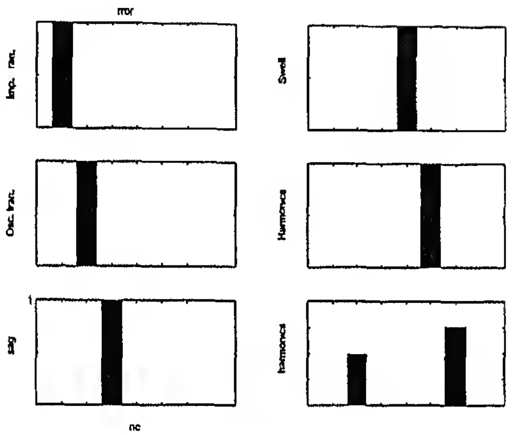


Figure A 1 Bayes classifier error matrix with feature range and individual class feature ranges are same as in Figure 5 3 but the data distribution is as shown in Figure A 14 We can observe that the Bayes classifier may give bad results when the data is very irregular Here we can observe in Figure A 14 that the data distribution of Interharmonics class for Dominant Interharmonic feature is distributed by towards the limits and no data at the center So, it is misclassified by the Oscillatory transients by about 40 % of the test data This is a case which is not an usual case in real power system

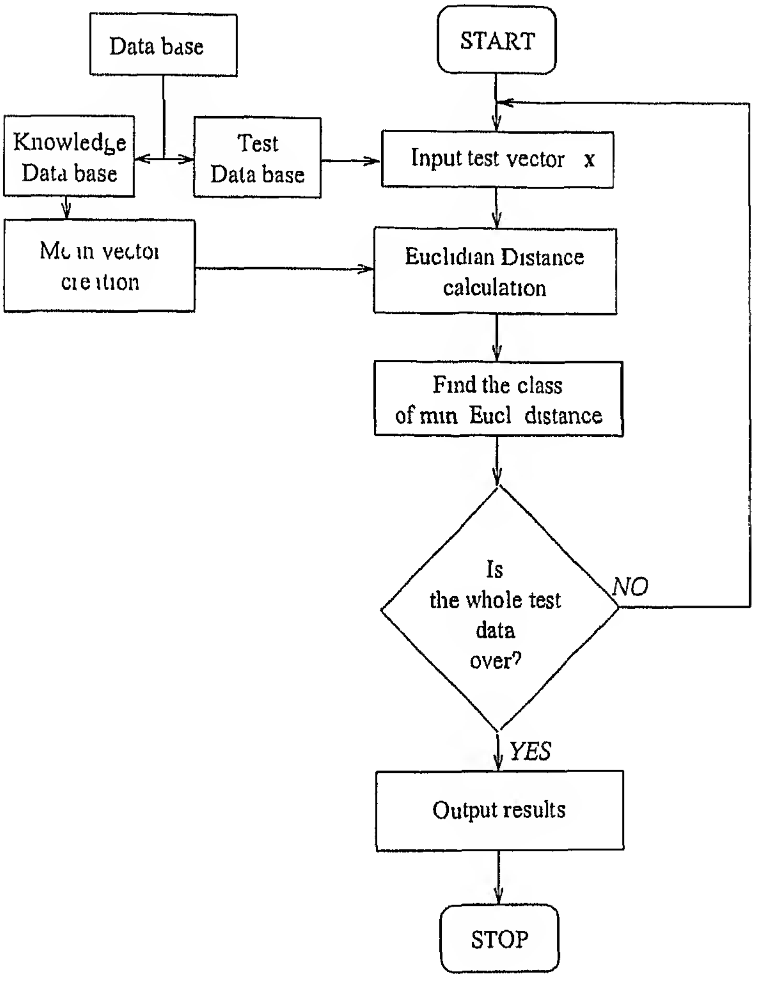


Figure A 2 Flow chart of testing Euclidean minimum distance classifier

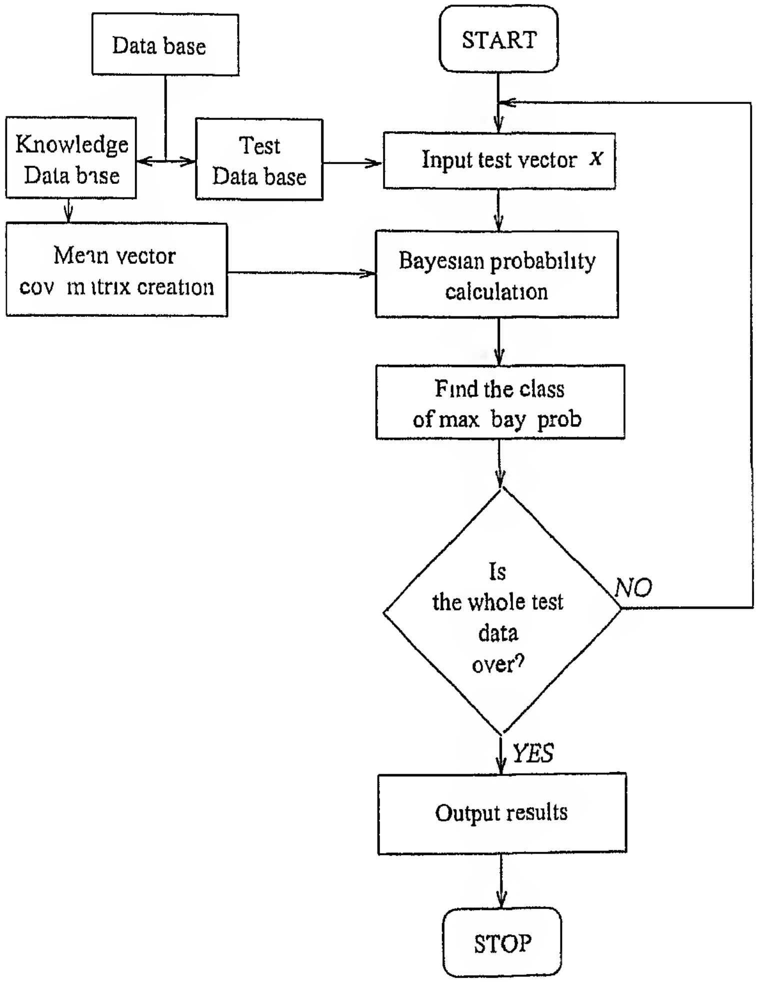


Figure A 3 Flow chart of testing Bayes classifier

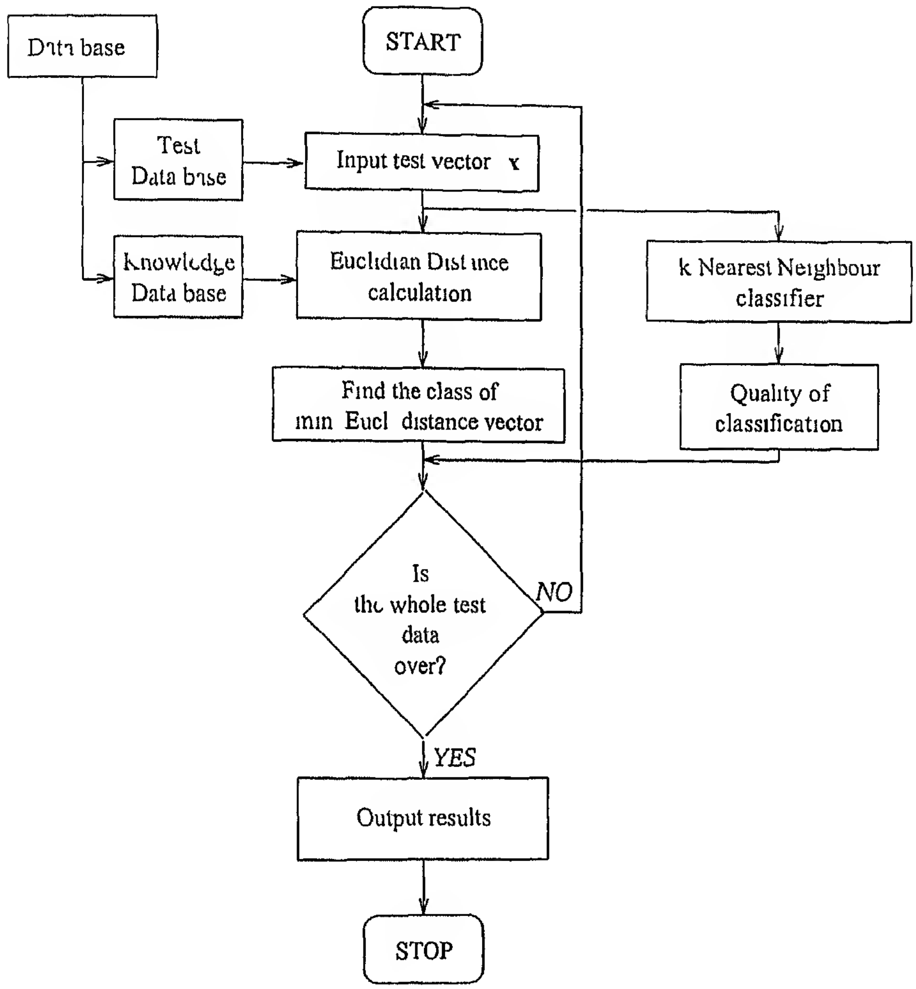


Figure A 4 Flow chart of testing k nearest neighbour classifier

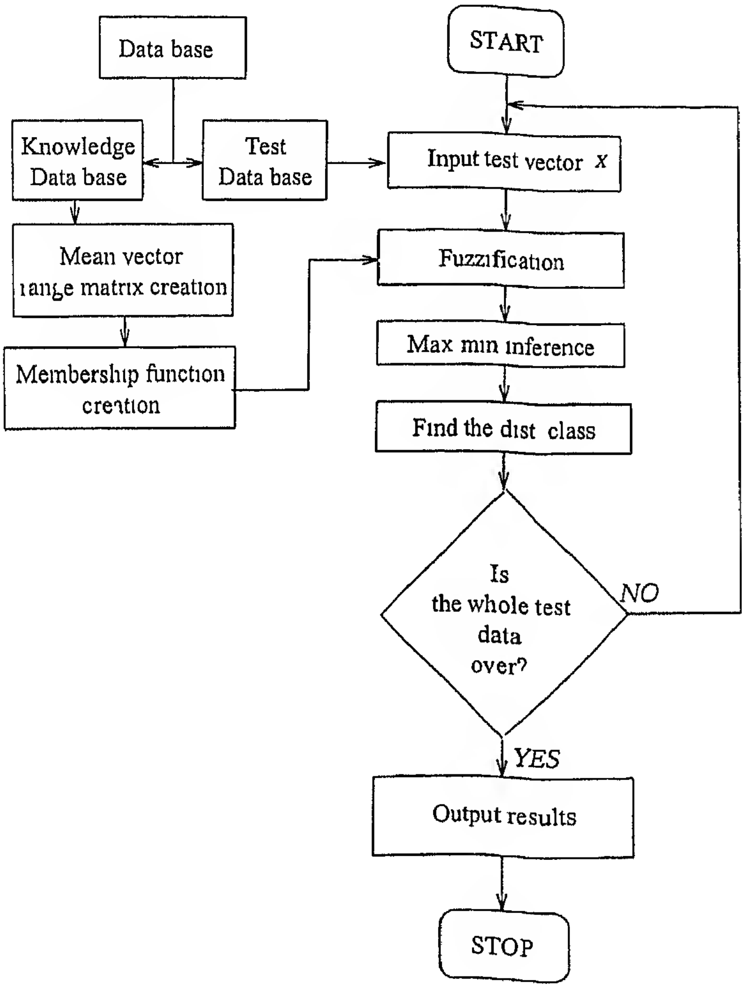


Figure A 5 Flow chart of testing Fuzzy pure classifier

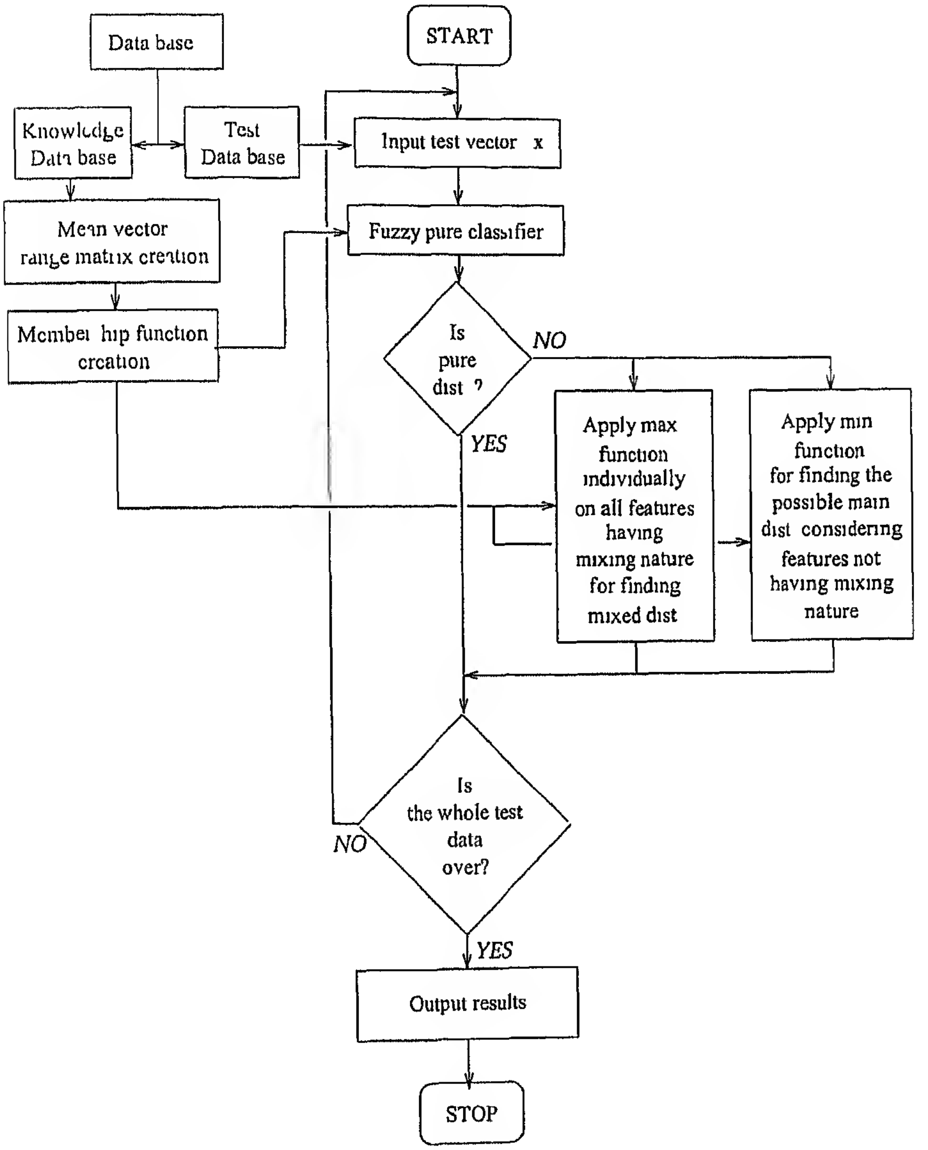


Figure A 6 Fuzzy sequential classifier flow chart

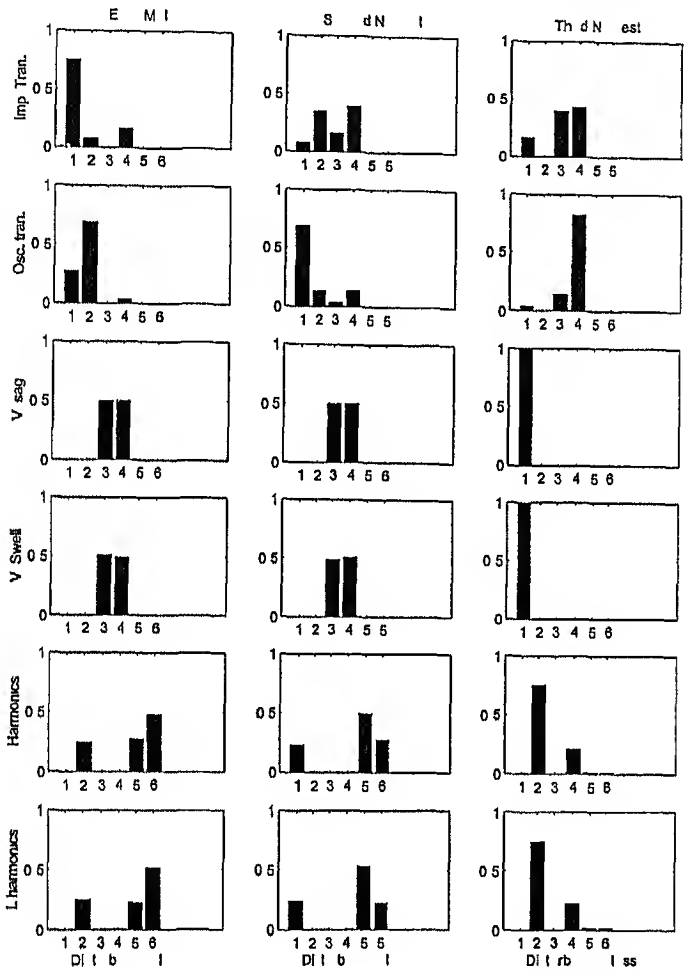


Figure A 7 Euclidean classifier error matrix when duration and rise time are not redistributed using logarithmic function

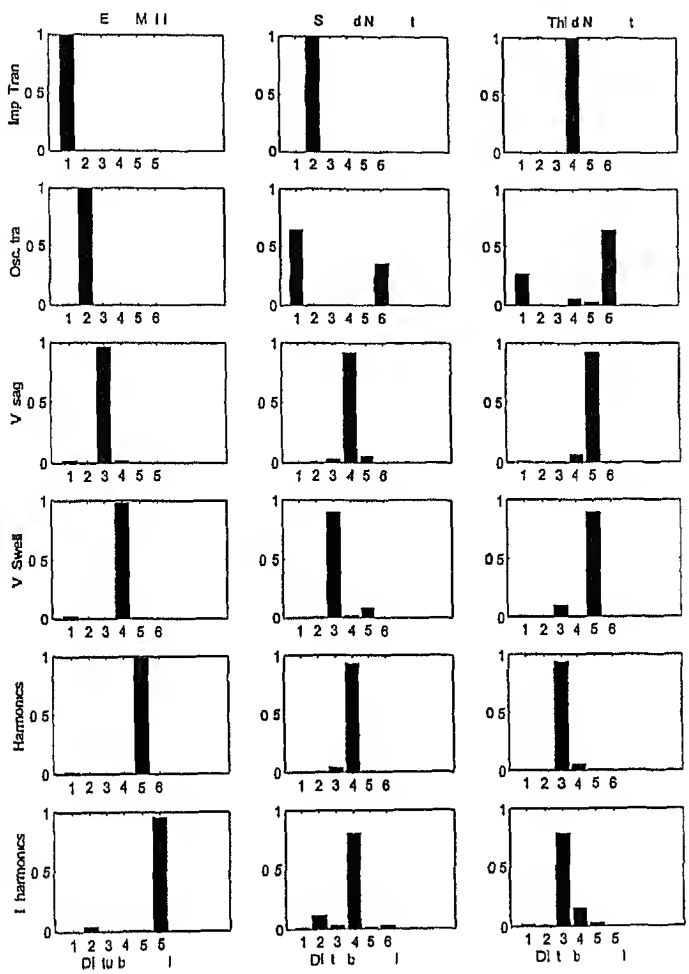


Figure A 8 Euclidean distance classifier error matrix with data is same as in Figure 5 3 except that the THD feature ranges for every disturbance is nonlinearly mapped to feature space The try is to equalise the range of different classes for the same feature Here the results are improved compared to Figure 5 3 for the class represents Harmonics Here, ie sectioning of THD feature reduces it s misclassification with neighbouring classes like voltage sag, voltage swell etc So, we can judge that the uniformity of the feature ranges of different classes for a particular feature is good to improve the classifier performance

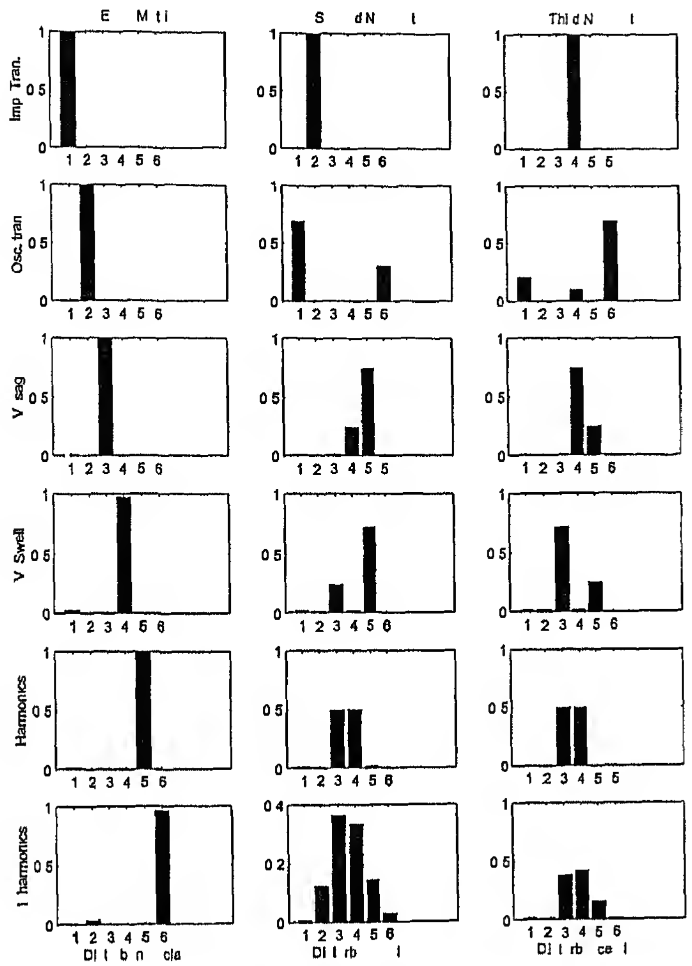


Figure A 9 Euclidean distance classifier error matrix with data is same as in Figure 5 3 except that peak voltage, RMS voltage and THD features are nonlinearly sectioned The sectioning aim is stated in Figure A 8 If we use re sectioning procedure for peak voltage and RMS voltage feature also then the results for the second and third class belongingness for the harmonics class improves Due to re sectioning, the distance from Harmonics mean vector to voltage sag and voltage swell mean vectors becomes equal As a results the second class and third class belongingness for harmonics for voltage sag and voltage swell becomes equal due to uniform distribution of data samples This case results differ with that in Figure A 8 Same results can be observed for Interharmonics class data samples results

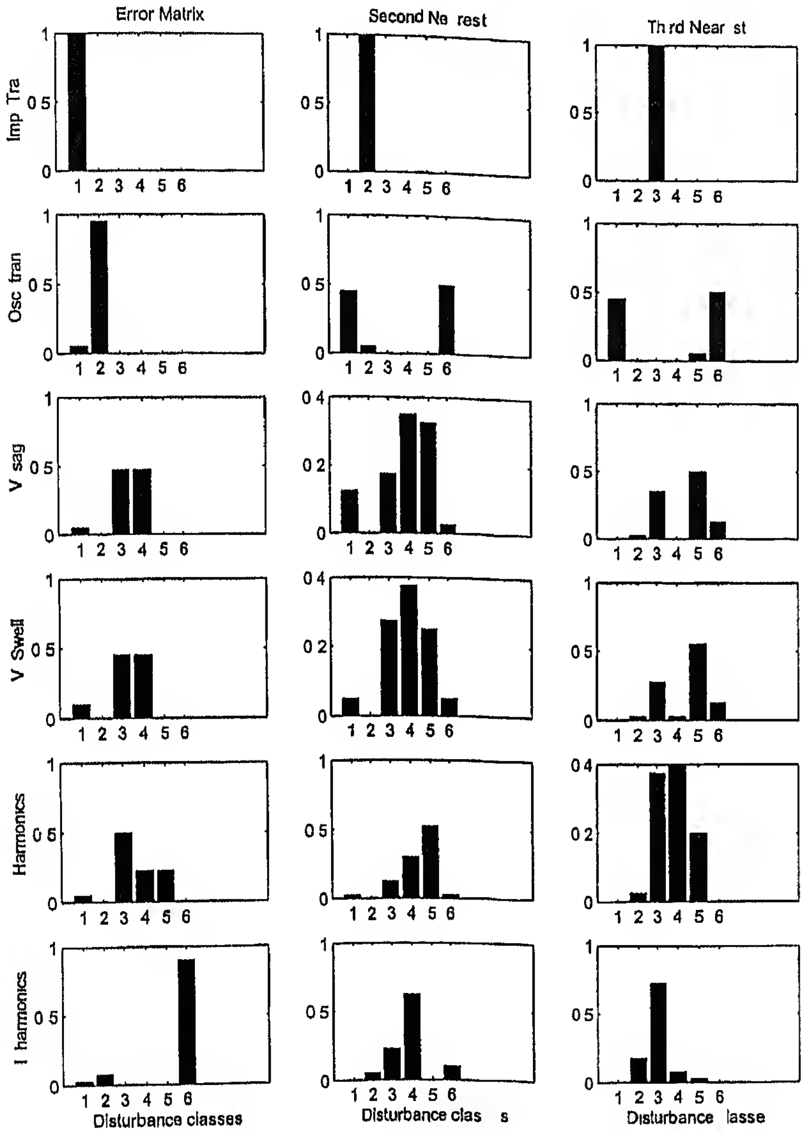


Figure A 10 Euclidean distance classifier error matrix with data number for every class, feature range and individual class feature ranges are same as in Figure 5 3 but the data is abnormally distributed as shown in Figure A 14 The results becomes less accurate due to shifting of mean vectors of the class from it s cluster center compared to results of uniformly distributed data as shown in Figure 5 3

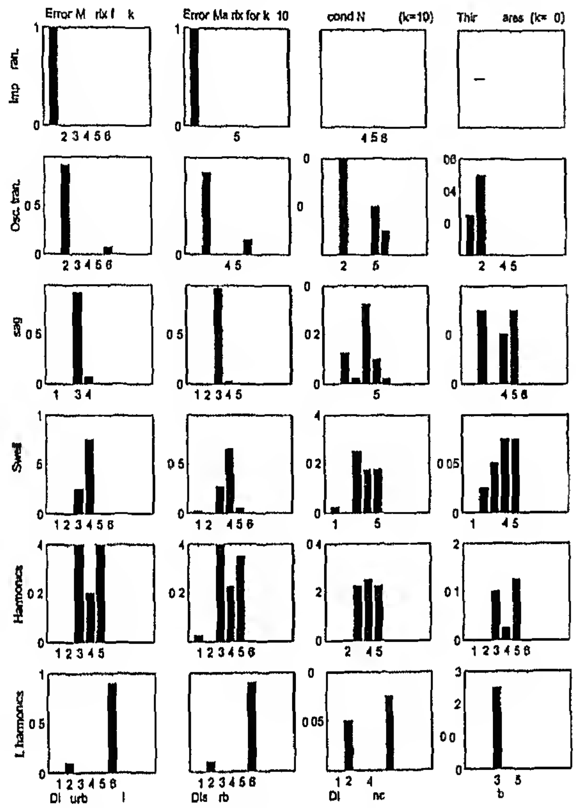


Figure A 11 k NN classifier error matrix with data distribution is same as in Figure A 10 The data samples for every class is 100 out of which 60 is for training and 40 for testing The first and second columns represent the error matrices for k=1 and k=10 nearest neighbour classifiers respectively The third and forth columns represent second nearest and third nearest class error matrices respectively for k=10 NN classifier The results are better than the results of Euclidean distance classifier for the same type of data as seen in Figure A 10 It is because the nonlinearity of the decision surface increases with decrease in nearest neighbours So, it functions as a piecewise linear classifier which can adapt to the abnormal distribution of the data in a better way compared to Euclidean distance classifier The results become less accurate compared to those with uniform distribution shown in Figure 5 5 due to decentralisation of data samples in every class cluster itself The randomness of test samples differs than that with training dataset in abnormally distributed data Results of l =1 and k=10 NN classifiers are almost same as seen here The more information about neighbouring classes can be available by this classifiers from third and fourth column results We can have a nice idea about the orientation of data vectors of different classes from this classifier results

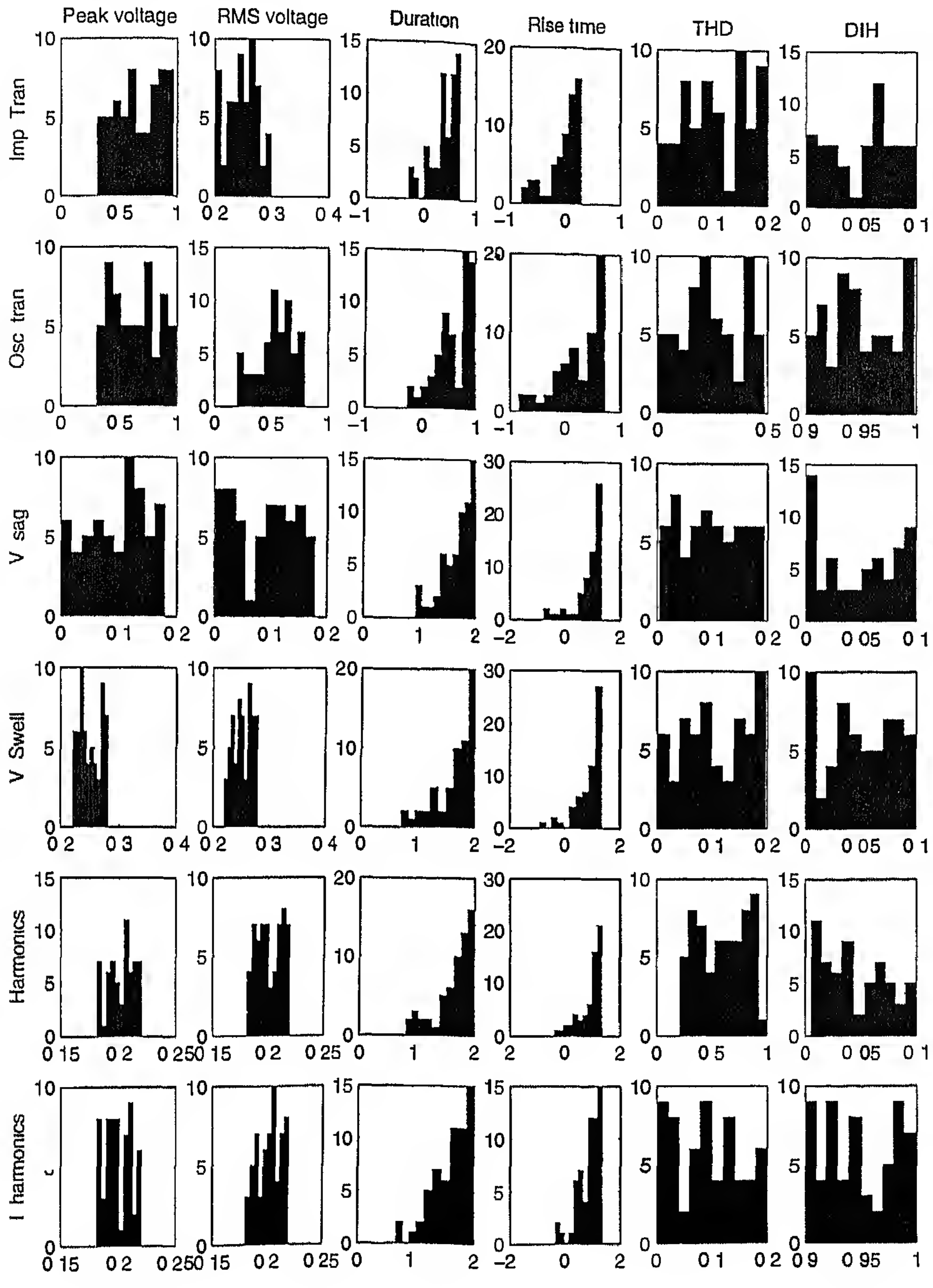


Figure A 12 Data distributed uniformally for 6 disturbances and 6 features and 100 data points for every disturbance The 100 data samples for every disturbance is shown Features peak voltage, RMS voltage, THD and Dominant Interharmonics are normalised to 1 per unit Duration and rise time features are generated first with 0 100 and 0 20 scales respectively and then $\log_{10}$ function is applied to redis tribute them So, they are not seen distributed as others though they are originally having same distributions Their scaling after redistribution is comparatively same as other features

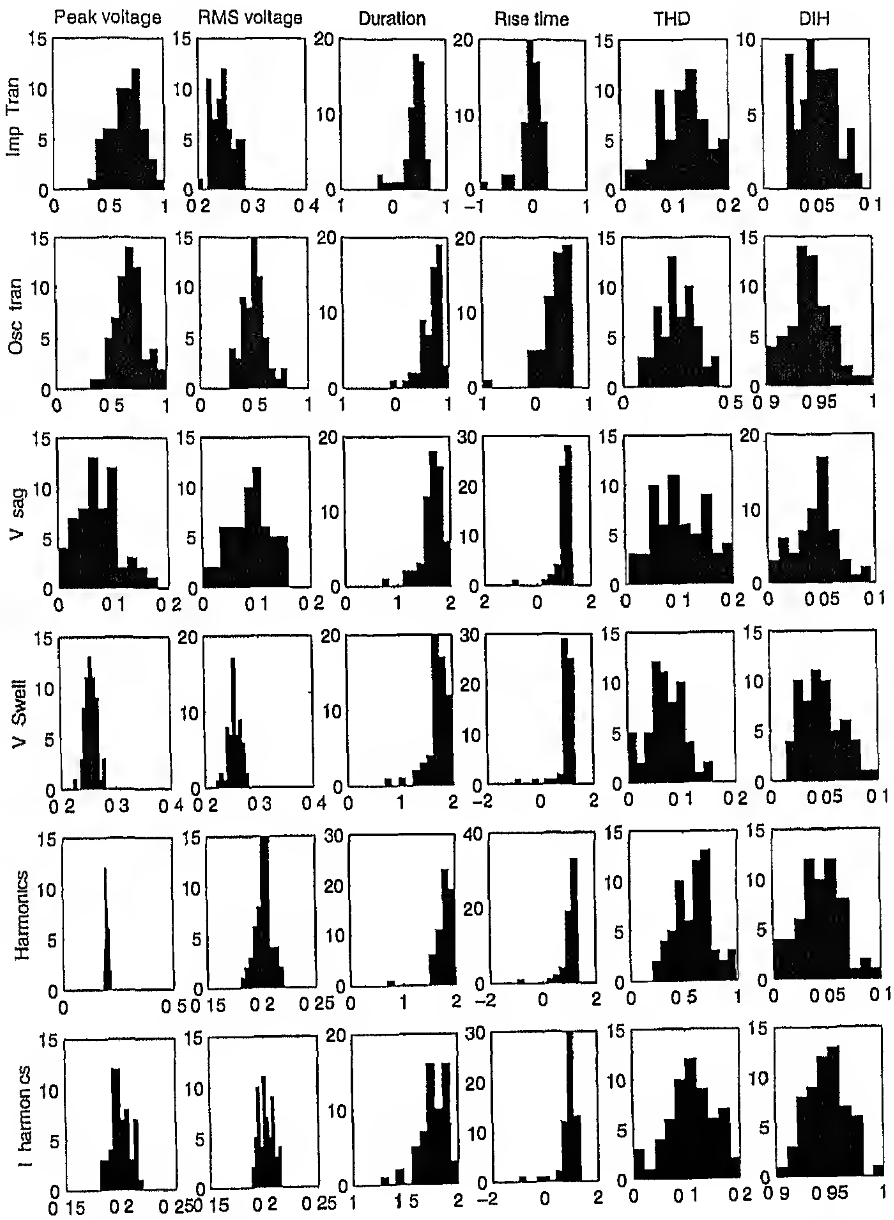


Figure A 13 Data is same as in Figure A 12 except that it is a Normal distribution

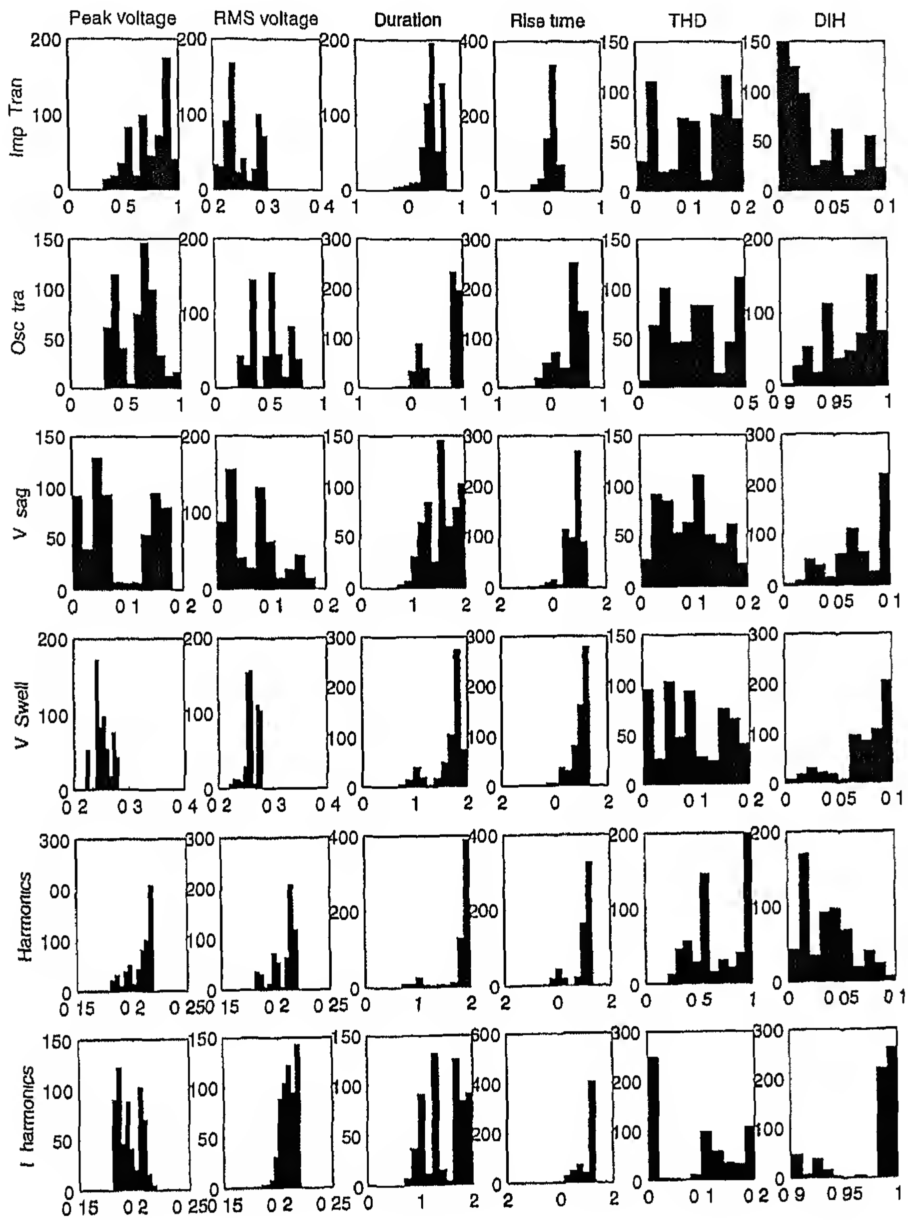


Figure A 14 Data is same as in Figure A 12 except that it is an abnormal distribution with 1000 data samples for every class It is different for different class feature ranges It is distributed randomly by sectioning the individual class feature range and data samples randomly and distributing those number of data samples in respective sub feature ranges As we can't assure about the perfect distribution of the real system disturbance data distribution for every feature, this may be one of the possible distribution for testing the classifiers

Notation

$\{0,1\}^n$	n dimensional binary hypercube
$[a,b]$	closed interval of real valued numbers between a and b
a^T	transpose of column vector a
$\mathrm{a}^T\mathrm{b}$	inner product of vector a and b
C^{-1}	inverse of matrix C
$\lvert \mathrm{C} \rvert$	Determinant of matrix C
$D(\mu_j, \mathrm{x})$	Euclidean distance between mean μ_j and vector x
$\lVert \mathrm{x} \rVert$	Euclidean norm of vector x
$\langle \, \rangle$	mathematical expectation or mean

Abbreviations

DIH	Dominant Interharmonics
GUI	Graphical User Interface
ln	natural logarithm
$\log_{10}$	base 10 logarithm
max	maximum
min	minimum
NNC	nearest neighbour classifier
NN	nearest neighbour
OR	OR Gate
PQ	power quality
RCE	restricted Coulomb energy
RMS	root mean square
THD	Total Harmonic Distortion

Symbols

AND	logical and operation
$\mathbf{C}$	autocorrelation matrix
c	i^{th} eigenvector of matrix C
$exp(a)$	exponential function
f	frequency
f_m	maximum frequency
f_i	normal frequency
H	error matrix
$\mathbf{I}$	identity matrix
$J(W)$	criterion objective function
k	number of classes
$mean_{i=1}^{N_j}(\mathbf{p}^j)$	mean vector of N_j vectors of j^{th} class
N	number of data samples
n	number of features
N_i	number of data samples in i^{th} class
NOT	logical not operation
OR	logical or operation
$P(i/\mathbf{x})$	conditional probability of unit i winning upon the presentation of $\mathbf{x}$
$\mathbf{R}$	space of n dimensional real valued numbers also used to designate Euclidean space
$S(\mathbf{x})$	classifier mapping function from feature space to decision space
T_e	Duration of Event
t_r	Rise time
V_m	Peak voltage
V_{RMS}	Voltage magnitude
$\mathbf{x}, \mathbf{x}_o$	input vector
x_i	i^{th} feature component value
Δf_m	small change in frequency
λ_i	i^{th} eigen value of matrix C
μ_q	mean vector of q^{th} class
μ	fuzzy membership function
$\mu_{ij}(x_j)$	membership value of x_j for i^{th} class
Ω	output class set
ω	output class
σ^2	variance
$\mathbf{K}$	co variance matrix
$\in$	symbol for belongs to

Bıblıography

[1] Dougheıty J , Stebbıns, W *Poweı qualıty a utılıty and ındustry perspec tıvc* Textıle Γıbre and Fılm Industry Technıcal Conference pp 5 10–5 20 1997

[2] Dudı R O *Patteın Recognıtıon for HCI* Sın Jose State Unıveısıty 1996 1997 *http //www engı sjsu edu/~knapp/IICIRODPR/PR_home htm*

[3] Dugın, R McCıunıghan, M , Beaty H *Electrıcal power system qualıty* McCıaw IIıll, New Yoık 1996

[4] I ussel, D *Self learnıng classıficatıon tree (SELECT) a human lıke ap pıoach to fault dıagnosıs* EUΓIT 97, fifth European Congress on Intellıgent Iechnıques and Soft Computıng Sept 8 11 Aachen Cermany

[5] I ussel, D *Learnıng fuzzy dıagnosıs system for supervısıon* Fıfth Interna tıonal Confeıence on Control Automıtıon Robotıcs and Vısıon 8 11 De cembeı 1998 Sıngıpoıe

[6] IIıssoun, M II *Γundamentals of Aıtıficıal Neural Networks*, Prentıce IIall of Indıa Pvt Ltd pp 97–98, 1995

[7] IIıssoun M II *Fundamentals of Artıficıal Neural Networks* Prentıce Hall of Indıı Pvt Ltd , pp 311–315 1995

[8] IIoekstrı, A , Duın, R P W *On the Nonlıneaıty of Pat teın Classıfieıs*, Delft Unıveısıty of Technology May 1996 *http //valhalla ph tn tudelft nl/generalızatıon/papers/ıcpr96/ıcpr96 html*

[9] Khıını K J , Yamany M S , Farag A A *Classıficatıon of the Effects of F actın Under Treatment of drugs ın Endothelıal Cells*, Unıversıty of Louısvılle Oct 1996 *http //www cvıp uofl edu/~yamany/annıe/annıe html*

[10] Leonhıardt, S , Ayoubı M *Methods of fault dıagnosıs*, Control Engg Prac tıce, Vol 5, No 5, pp 683–692, 1997

[11] Mıtlıb Veısıon 5 2 *Usıng Matlab* The Math Works, Inc , 1997

[12] Ridder De D *Shared weights neural networks in image analysis* M Sc Thesis Delft University of Technology March 1996 *http //www ph tn tudelft nl/Research/neural/feature extraction/papers/thesis/thesis html*

[13] Schuermann J *Pattern classification a unified view of statical and neural approaches* John Wiley & Sons Inc New York 1996 Pattern Recognition for IICI

[14] Theodoridis S Koutroumbas K *Pattern Recognition* Academic Press San Diego pp 186–187 1999

A 130812

Date Slip

This book is to be returned on the date last stamped

A130812